

# Claris FileMaker Server 19

Guide de la Publication Web personnalisée



© 2004–2020 Claris International Inc. Tous droits réservés.

Claris International Inc.  
5201 Patrick Henry Drive  
Santa Clara, Californie 95054

FileMaker, FileMaker Cloud, FileMaker Go et le logo en forme de dossier sont des marques commerciales de Claris International Inc. déposées aux États-Unis et dans d'autres pays. Claris, le logo Claris, Claris Connect et FileMaker WebDirect sont des marques commerciales de Claris International Inc. Toutes les autres marques commerciales sont la propriété de leurs détenteurs respectifs.

La documentation produit FileMaker est protégée par la législation sur les droits d'auteur. Vous n'êtes pas autorisé à créer des copies supplémentaires ni à distribuer cette documentation sans l'accord écrit de Claris. Vous devez posséder une copie sous licence valide de FileMaker pour utiliser cette documentation.

Toutes les personnes, sociétés, adresses email et URL citées dans les exemples sont fictives et toute ressemblance avec des personnes, des sociétés, des adresses email ou des URL existantes ne serait que pure coïncidence. La liste des auteurs des produits est disponible dans les documents Remerciements fournis avec ce logiciel. La liste des auteurs de la documentation est disponible dans [Remerciements Documentation](#). Les produits tiers et les adresses URL sont mentionnés à titre indicatif uniquement, et non pas à titre de recommandation. Claris International Inc. se dégage de toute responsabilité concernant les performances de ces produits.

Pour plus d'informations, consultez notre [site Web](#).

Edition : 01

# Table des matières

<b>Préface</b>	8
À propos de ce guide	8
Emplacement de la documentation produit FileMaker	8
<b>Chapitre 1</b>	
<b><i>Introduction à la Publication Web personnalisée</i></b>	9
A propos du moteur de publication Web	10
Traitement d'une requête du moteur de publication Web	10
Publication Web personnalisée avec XML	11
Publication Web personnalisée avec PHP	11
Comparaison entre XML et PHP	11
Raisons de choisir XML	11
Raisons de choisir PHP	12
<b>Chapitre 2</b>	
<b><i>Préparation des bases de données pour la Publication Web personnalisée</i></b>	13
Activation de la Publication Web personnalisée dans une base de données	13
Accès à une base de données protégée	14
Protection de vos bases de données publiées	15
Prise en charge du serveur Web pour les types de média Internet (MIME)	16
À propos de la publication du contenu des rubriques Conteneur sur le Web	16
Objets de rubriques Conteneur imbriqués dans une base de données	16
Rubriques Conteneur avec des références de fichier stockées	16
Types de données de conteneur avec données stockées en externe	17
Rubriques Conteneur et téléchargement progressif	18
Affichage des données d'une rubrique Conteneur par les utilisateurs Web	18
Scripts FileMaker et Publication Web personnalisée	18
Astuces et considérations à propos des scripts	19
Comportement des scripts dans les solutions de Publication Web personnalisée	20
Déclencheurs de scripts et solutions de Publication Web personnalisée	20
<b>Chapitre 3</b>	
<b><i>À propos de la fonction de Publication Web personnalisée utilisant le langage XML</i></b>	21
Création de sites Web dynamiques avec le moteur de publication Web	21
Fonctions clés de la Publication Web personnalisée avec XML	22
Configuration pour la publication Web	22
Configuration requise pour la publication d'une base de données à l'aide de la Publication Web personnalisée	22
Éléments devant être employés par les utilisateurs Web pour accéder à une solution de publication Web personnalisée	23
Connexion à Internet ou à un intranet	23
Étape suivante	23

## Chapitre 4

<b>Accès aux données XML avec le moteur de publication Web</b>	24
Utilisation de la Publication Web personnalisée avec XML	24
Comparaison du moteur de publication Web avec l'importation et l'exportation XML	24
Génération de données XML par le moteur de publication Web à partir d'une requête	25
Processus général d'accès aux données XML à partir du moteur de publication Web	26
À propos de la syntaxe d'URL pour les données XML et les objets de type Conteneur	27
À propos de la syntaxe d'URL pour les données XML	27
À propos de la syntaxe d'URL pour les objets FileMaker Pro de type Conteneur dans les solutions XML	27
À propos du codage de texte URL	29
Accès aux données XML via le moteur de publication Web	29
À propos des espaces de nom pour le langage XML de FileMaker	30
À propos des codes d'erreurs des bases de données FileMaker Pro	30
Récupération des définitions de type de document pour les grammaires FileMaker	30
Utilisation de la grammaire fmresultset	30
Description des éléments de la grammaire fmresultset	31
Données XML dans la grammaire fmresultset	33
Utilisation d'autres grammaires XML FileMaker	34
Description des éléments de la grammaire FMPXMLRESULT	34
Données XML dans la grammaire FMPXMLRESULT	35
Description des éléments dans la grammaire FMPXMLLAYOUT	36
Données XML dans la grammaire FMPXMLLAYOUT	38
À propos des données UTF-8	39
Utilisation de chaînes de requête FileMaker pour rechercher des données XML	39
Passage d'un modèle à un autre pour obtenir une réponse XML	41
Comprendre le traitement d'une requête XML	42
Dépannage de l'accès aux documents XML	42

## Chapitre 5

<b>Noms corrects utilisés dans les chaînes de requête XML</b>	43
À propos des commandes et des paramètres de requête	43
Consignes d'utilisation des commandes et des paramètres de requête	44
Analyse de commande de requête	45
À propos de la syntaxe d'un nom de rubrique entièrement qualifié	46
Utilisation de commandes de requêtes avec des rubriques de table externe	46
À propos de la syntaxe de spécification d'une rubrique de type Global	48
Références des commandes de requête	48
Commande de requête –dbnames (noms des bases de données)	49
Commande de requête –delete (supprimer l'enregistrement)	49
Commande de requête –dup (dupliquer l'enregistrement)	49
Commande de requête –edit (modifier l'enregistrement)	49
Commandes de requête –find, –findall ou –findany (rechercher les enregistrements)	50
Commande de requête –findquery (recherche composée)	50
Commande de requête –layoutnames (noms des modèles)	51
Commande de requête –new (nouvel enregistrement)	51
Commande de requête –scriptnames (noms des scripts)	52
Commande de requête –view (affichage des informations de modèle)	52

Références de paramètre de requête	52
Paramètre de requête –db (nom de base de données)	52
Paramètre de requête –delete.related (suppression d'enregistrements de la table externe)	53
Paramètre de requête –field (nom de rubrique Conteneur)	53
Paramètre de requête fieldname (nom de rubrique non-Conteneur)	53
Paramètre de requête fieldname.op (opérateur de comparaison)	54
Paramètre de requête –lay (modèle)	55
Paramètre de requête –lay.response (changer de modèle pour la réponse)	55
Paramètre de requête –lop (opérateur logique)	56
Paramètres de requête –max (nombre maximal d'enregistrements)	56
Paramètre de requête –modid (ID de la modification)	56
Paramètre de requête –query (requête de recherche composée)	57
Paramètre de requête –recid (ID d'enregistrement)	58
Paramètre de requête –relatedsets.filter (filtrage d'enregistrements de table externe)	58
Paramètre de requête –relatedsets.max (limitation d'enregistrements de table externe)	59
Paramètre de requête –script (Script)	60
Paramètre de requête –script.param (transfert du paramètre au script)	60
Paramètre de requête –script.prefind (script avant une recherche)	60
Paramètre de requête –script.prefind.param (transfert du paramètre au script avant recherche)	61
Paramètre de requête –script.presort (script avant un tri)	61
Paramètre de requête –script.presort.param (transfert du paramètre au script avant recherche)	61
Paramètre de requête –skip (enregistrements ignorés)	62
Paramètre de requête –sortfield (rubrique de tri)	62
Paramètre de requête –sortorder (ordre de tri)	63

## Chapitre 6

<b>À propos de la fonction de Publication Web personnalisée utilisant le langage PHP</b>	64
Fonctions clés de la Publication Web personnalisée avec PHP	64
Configuration de la Publication Web personnalisée	64
Configuration requise pour la publication d'une base de données à l'aide de la Publication Web personnalisée	65
Éléments devant être employés par les utilisateurs Web pour accéder à une solution de publication Web personnalisée	65
Connexion à Internet ou à un intranet	66
Installation manuelle de l'API FileMaker pour PHP	66
Étape suivante	67

## Chapitre 7

<b>Présentation de la Publication Web personnalisée avec PHP</b>	68
Fonctionnement du moteur de publication Web avec les solutions PHP	68
Étapes générales de la Publication Web personnalisée avec PHP	68

## Chapitre 8

### *Utilisation de l'API FileMaker pour PHP*

<i>Utilisation de l'API FileMaker pour PHP</i>	71
Où trouver des informations supplémentaires	71
Référence API FileMaker pour PHP	71
Prise en charge API FileMaker pour PHP	71
Utilisation de la classe FileMaker	72
Objets de classe FileMaker	72
Objets de commande FileMaker	72
Décodage des données à utiliser dans l'API FileMaker	73
Connexion à une base de données FileMaker Pro	73
Utilisation des enregistrements	74
Création d'un enregistrement	74
Duplication d'un enregistrement	75
Edition d'un enregistrement	75
Suppression d'un enregistrement	76
Exécution de scripts FileMaker	76
Obtention de la liste des scripts disponibles	76
Exécution d'un script FileMaker	77
Exécution d'un script avant l'exécution d'une commande	77
Exécution d'un script avant le tri d'un ensemble de résultats	77
Exécution d'un script après la génération d'un ensemble de résultats	77
Ordre d'exécution du script	78
Utilisation des modèles FileMaker Pro	79
Utilisation des tables externes	80
Création d'une liste des tables externes définies sur un modèle spécifique	80
Obtention des noms de tables externes d'un objet de résultat spécifique	80
Obtention d'informations sur les tables externes d'un modèle spécifique	80
Obtention d'informations sur une table externe spécifique	80
Obtention du nom de table d'une table externe	81
Obtention des enregistrements externes d'un enregistrement spécifique	81
Création d'un nouvel enregistrement dans une table externe	81
Suppression d'un enregistrement d'une table externe	81
Utilisation des listes de valeurs	82
Obtention des noms de toutes les listes de valeurs d'un modèle spécifique	82
Obtention d'un tableau de toutes les listes de valeurs d'un modèle spécifique	82
Obtention des valeurs pour une liste de valeurs nommée	82
Exécution de requêtes	83
Utilisation de la commande Rechercher tout	84
Utilisation de la commande Rechercher au moins	84
Utilisation de la commande Rechercher	85
Utilisation d'une commande Recherche composée	85
Traitement des enregistrements d'un ensemble de résultats	87
Limitation du nombre de lignes des tables externes renvoyées par les requêtes	88
Prévalidation des commandes, des enregistrements et des rubriques	88
Prévalidation des enregistrements dans une commande	90
Prévalidation des enregistrements	90
Prévalidation des rubriques	90
Traitement des erreurs de validation	90
Gestion des erreurs	92

## Chapitre 9

### ***Stockage, test et suivi d'un site***

Stockage d'un site de Publication Web personnalisée	93
Test d'un site de Publication Web personnalisée	94
Feuilles de style permettant de tester les sorties XML	95
Suivi de votre site	96
Utilisation des journaux d'erreurs et d'accès au serveur Web	96
Utilisation du journal du moteur de publication Web	96
Utilisation du journal des erreurs du module de serveur Web	98
Utilisation des journaux Tomcat	98

## Annexe A

### ***Codes d'erreur de la Publication Web personnalisée***

Numéros de code d'erreur au format XML	99
Numéros des codes d'erreur des bases de données FileMaker Pro	100

## ***Index***

101

# Préface

## À propos de ce guide

Ce guide part de l'hypothèse que vous avez déjà une certaine expérience dans l'utilisation de Claris™ FileMaker® Pro pour la création de bases de données. Vous devez comprendre les bases de la conception des bases de données avec FileMaker Pro, ainsi que les concepts de rubriques, de liens, de modèles, de tables externes et d'objets Conteneur. Pour en savoir plus sur FileMaker Pro, consultez l'[Aide FileMaker Pro](#).

Ce guide part également du principe que vous avez déjà une certaine expérience dans le développement de sites Web, et notamment dans l'utilisation de technologies comme XML ou PHP, afin d'intégrer les données FileMaker aux sites et applications Web.

Concernant la Publication Web personnalisée sous Claris FileMaker Server, ce guide donne des informations sur :

- les éléments nécessaires au développement d'une solution de Publication Web personnalisée ;
- la façon de publier des bases de données à l'aide du langage XML ;
- la façon d'obtenir des données XML à partir de bases de données hébergées par FileMaker Server ;
- la façon de publier des bases de données à l'aide du langage PHP ;
- la façon d'utiliser l'API FileMaker pour PHP à partir de bases de données hébergées par FileMaker Server ;
- les éléments que les utilisateurs Web doivent employer pour accéder à une solution de publication Web personnalisée.

## Emplacement de la documentation produit FileMaker

- L'Aide FileMaker Server est disponible sur chaque page de l'Admin Console de FileMaker Server. Faites défiler jusqu'au bas de la page, puis cliquez sur **Aide**.
- La documentation FileMaker Server est disponible sur chaque page de l'Admin Console. Faites défiler jusqu'au bas de la page, puis cliquez sur **Documentation**.
- Sur le Web, accédez au [Centre de documentation produit](#).



# Chapitre 1

## Introduction à la Publication Web personnalisée

FileMaker Server vous permet de publier vos bases de données FileMaker Pro sur Internet ou sur un intranet. Vous pouvez procéder de différentes manières :

**FileMaker WebDirect** : Avec FileMaker WebDirect™ de Claris, vous avez la possibilité de publier rapidement et simplement des modèles à partir d'une base de données sur le Web. Aucune installation de logiciels supplémentaires n'est nécessaire. Avec un navigateur Web compatible et un accès à Internet ou à un intranet, les utilisateurs Web peuvent se connecter à votre solution FileMaker WebDirect pour consulter, éditer, trier ou rechercher des enregistrements, à condition que vous leur donniez des privilèges d'accès.

Avec FileMaker WebDirect, FileMaker Server doit être en cours d'exécution sur l'ordinateur hôte. L'interface utilisateur ressemble à la version bureau de l'application FileMaker Pro. Les pages et les formulaires Web sur lesquels l'utilisateur Web interagit dépendent des modèles et des affichages définis dans la base de données FileMaker Pro. Consultez le [Guide de FileMaker WebDirect](#).

**Publication statique** : Si vos données ne changent pas souvent ou si vous ne souhaitez pas que les utilisateurs puissent se connecter directement à votre base de données, vous pouvez faire appel à la publication statique. Avec cette dernière, vous exportez les données d'une base de données FileMaker Pro pour créer une page Web, que vous pouvez ensuite personnaliser à l'aide du langage HTML. Ainsi, la page Web ne change pas lorsque des données de la base sont modifiées et que les utilisateurs ne se connectent pas directement à votre base de données. (Avec FileMaker WebDirect, les données sont mises à jour dans le navigateur Web dès qu'elles sont mises à jour dans la base de données). Consultez l'[Aide FileMaker Pro](#).

**FileMaker Data API** : Pour ceux aguerris aux techniques de l'architecture REST (Representational State Transfer), la plateforme FileMaker de Claris fournit une mise en œuvre d'API REST qui permet aux services Web d'accéder aux données dans les solutions hébergées. Votre service Web appelle FileMaker Data API pour obtenir un jeton d'authentification et accéder à une solution hébergée. Il utilise ensuite ce jeton dans les appels suivants pour créer des enregistrements, les mettre à jour, les supprimer et effectuer des recherches. FileMaker Data API renvoie des données au format JSON (JavaScript Object Notation). Consultez le [Guide de FileMaker Data API](#).

**Publication Web personnalisée** : Pour intégrer votre base de données FileMaker Pro à un site Web personnalisé, utilisez les technologies de Publication Web personnalisée disponibles avec FileMaker Server. Pour FileMaker Server, qui héberge les bases de données publiées, il n'est pas nécessaire d'installer ni d'exécuter FileMaker Pro pour que la Publication Web personnalisée soit disponible.

La Publication Web personnalisée vous permet :

- intégrer votre base de données à un autre site Web ;
- déterminer la façon dont les utilisateurs interagissent avec les données ;
- contrôler l'affichage des données dans les navigateurs Web.

FileMaker Server propose deux technologies de Publication Web personnalisée :

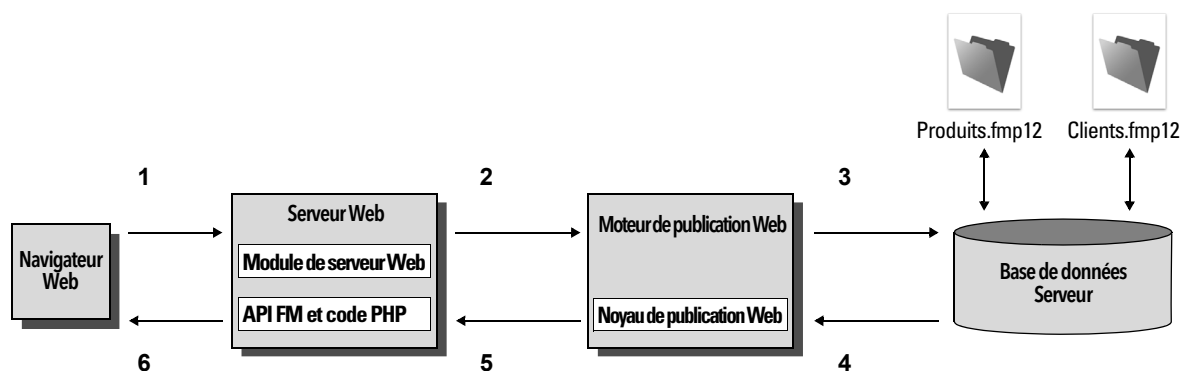
- Publication Web personnalisée avec XML : Utilisez la publication de données XML pour échanger des données FileMaker avec d'autres sites Web et applications. Grâce aux requêtes URL HTTP avec les commandes et les paramètres de requête de FileMaker, vous pouvez interroger une base de données hébergée par FileMaker Server et télécharger les données obtenues au format XML, et utiliser les données obtenues de la façon qui vous plaît.
- Publication Web personnalisée avec PHP : Utilisez l'API FileMaker pour PHP. Elle propose une interface PHP orientée objet pour les bases de données FileMaker Pro, ce qui permet d'intégrer vos données FileMaker dans une application Web PHP. Comme vous codez les pages Web PHP vous-même, vous avez un contrôle total de l'interface utilisateur et du mode d'interaction entre l'utilisateur et les données.

## A propos du moteur de publication Web

Pour prendre en charge FileMaker WebDirect et la Publication Web personnalisée, FileMaker Server utilise un jeu de composants logiciels appelé *Moteur de Publication Web de FileMaker Server*. Le moteur de Publication Web gère les interactions entre le navigateur Web d'un utilisateur, votre serveur Web et FileMaker Server.

**Publication Web personnalisée avec XML :** Les utilisateurs Web accèdent à votre solution de Publication Web personnalisée en cliquant sur un lien HREF ou en saisissant une URL (Uniform Resource Locator) qui indique l'adresse du serveur Web, ainsi qu'une requête qui contient la chaîne de requête FileMaker. Le moteur de publication Web renvoie les données XML spécifiées dans la requête de chaîne.

**Publication Web personnalisée avec PHP :** Lorsqu'un utilisateur Web accède à votre solution de Publication Web personnalisée, PHP sur FileMaker Server se connecte au moteur de Publication Web et répond via l'API de FileMaker pour PHP.



Utilisation du moteur de publication Web de FileMaker Server pour la Publication Web personnalisée

## Traitement d'une requête du moteur de publication Web

1. Une requête est envoyée au serveur Web par un navigateur Web ou une application.
2. Le serveur Web achemine la requête par le module serveur Web FileMaker jusqu'au moteur de publication Web.
3. Le moteur de publication Web recherche les données dans la base de données hébergée par le serveur de bases de données.

4. FileMaker Server envoie les données FileMaker demandées au moteur de publication Web.
5. Le moteur de publication Web convertit les données FileMaker pour répondre à la requête.
  - Pour les requêtes PHP, l'API FileMaker pour PHP convertit la requête PHP en requête XML. Le moteur de publication Web traite la requête XML puis renvoie les données XML à l'API FileMaker pour PHP. L'API FileMaker pour PHP convertit ensuite les données XML en objets PHP que l'application PHP peut utiliser.
  - Pour les requêtes XML, le moteur de publication Web envoie directement les données XML au serveur Web.
6. Le serveur Web envoie le résultat au navigateur Web ou au programme qui en fait la demande.

**Important** La sécurité est un aspect important de la publication de données sur le Web. Revoyez les consignes de sécurité dans le [Guide de sécurité FileMaker](#).

Pour plus d'informations sur la façon de préparer une base de données pour une Publication Web personnalisée, consultez le chapitre 2, « Préparation des bases de données pour la Publication Web personnalisée ».

## Publication Web personnalisée avec XML

La Publication Web personnalisée avec XML vous permet d'envoyer des demandes de requêtes à une base de données FileMaker Pro hébergée par FileMaker Server, puis d'afficher, de modifier ou de manipuler les résultats. L'utilisation d'une requête HTTP avec les commandes et les paramètres appropriés vous aide à récupérer les données FileMaker sous la forme d'un document XML. Vous pouvez alors exporter les données XML vers d'autres applications.

## Publication Web personnalisée avec PHP

L'API FileMaker pour PHP propose une interface PHP orientée objet aux bases de données FileMaker Pro. L'API de FileMaker pour PHP permet l'accès aux données et logiques stockées dans une base de données FileMaker Pro et leur publication sur le Web, ainsi que leur exportation vers d'autres applications. L'API gère également les commandes de recherche complexes et composées permettant l'extraction et le filtrage des informations stockées dans des bases de données FileMaker Pro.

Conçu à l'origine comme un langage de programmation procédural, PHP est devenu un langage de développement Web orienté objet. PHP propose des fonctionnalités de langage de programmation permettant de construire n'importe quel type de logique dans une page de site. Par exemple, vous pouvez utiliser les constructions logiques conditionnelles pour contrôler la génération de page, l'acheminement des données ou un déroulement d'opérations. PHP permet également l'administration du site et la sécurité.

## Comparaison entre XML et PHP

La section suivante propose quelques grandes lignes permettant de déterminer la meilleure solution pour votre site.

### Raisons de choisir XML

- La syntaxe des paramètres d'interrogation XML de FileMaker est conçue pour l'interaction avec les bases de données, ce qui simplifie le développement de solutions.

- XML est une norme W3C.
- XML est un format lisible par les hommes et les machines. Il gère Unicode, ce qui permet la communication des données dans n'importe quelle langue écrite.
- XML est bien adapté à la présentation d'enregistrements, de listes et de données structurées en liste.
- Vous pouvez utiliser FMPXMLRESULT pour accéder aux données XML en utilisant la Publication Web personnalisée et pour l'export XML à partir de bases de données FileMaker Pro.

**Remarque** Pour plus d'informations sur la Publication Web personnalisée avec XML, consultez le chapitre 3, « À propos de la fonction de Publication Web personnalisée utilisant le langage XML ».

### Raisons de choisir PHP

- PHP est un langage de script procédural orienté objet plus puissant, mais son apprentissage est relativement simple. Des nombreuses ressources sont disponibles pour la formation, le développement et l'assistance.
- L'API de FileMaker pour PHP permet l'accès aux données et logiques stockées dans une base de données FileMaker Pro et leur publication sur le Web, ainsi que leur export vers d'autres applications.
- PHP permet d'utiliser des logiques conditionnelles pour contrôler la construction de la page ou les flux.
- PHP propose des fonctionnalités de langage de programmation permettant de construire n'importe quel type de logique dans une page de site.
- PHP est l'un des langages de script les plus populaires du Web.
- PHP est un langage source ouvert disponible sur le site [php.net](http://php.net).
- PHP permet l'accès à une grande variété de modules tiers que vous pouvez intégrer dans vos solutions.

**Remarque** Pour plus d'informations sur la Publication Web personnalisée avec PHP, consultez le chapitre 6, « À propos de la fonction de Publication Web personnalisée utilisant le langage PHP ».

# Chapitre 2

## Préparation des bases de données pour la Publication Web personnalisée

Avant d'utiliser la Publication Web personnalisée avec une base de données, vous devez préparer la base de données et la protéger contre les accès non autorisés.

### Activation de la Publication Web personnalisée dans une base de données

Vous devez activer un privilège d'accès étendu pour la Publication Web personnalisée dans chaque base de données que vous souhaitez publier. Si vous n'activez pas de privilège d'accès étendu pour la Publication Web personnalisée dans la base de données, les utilisateurs Web ne seront pas en mesure de l'utiliser pour accéder à la base de données, même si elle est hébergée par un serveur FileMaker Server configuré pour prendre en charge un Moteur de publication Web.

#### Pour activer la Publication Web personnalisée pour une base de données :

1. Dans FileMaker Pro, ouvrez la base de données que vous souhaitez publier en utilisant un compte bénéficiant du jeu de privilèges Accès intégral. Une autre possibilité consiste à ouvrir la base de données en utilisant un compte bénéficiant des privilèges d'accès Gérer les autorisations étendues.
2. Affectez le privilège d'accès étendu pour la Publication Web personnalisée que vous souhaitez utiliser :
  - Pour la Publication Web personnalisée avec XML, utilisez fmxml
  - Pour la Publication Web personnalisée avec PHP, utilisez fmphp
3. Affectez un jeu de privilèges comprenant le privilège étendu Publication Web personnalisée à un ou plusieurs comptes, ou au compte Admin ou Invité.

**Remarque** Au moment de définir les noms des comptes et les mots de passe relatifs aux solutions de publication Web personnalisée, utilisez les caractères ASCII imprimables, par exemple **a-z**, **A-Z** et **0-9**. Pour sécuriser davantage encore les noms des comptes et les mots de passe, intégrez des caractères de ponctuation du type « ! » et « % » mais n'utilisez pas les deux points. Pour obtenir des informations sur la configuration des comptes, consultez l'[Aide FileMaker Pro](#).

## Accès à une base de données protégée

La Publication Web personnalisée vous permet de limiter l'accès à vos bases de données publiées à l'aide d'une protection par mot de passe de vos bases de données, d'un cryptage de la base de données et de connexions sécurisées. Lorsqu'ils utilisent une solution de Publication Web personnalisée pour accéder à une base de données, les utilisateurs Web peuvent être invités à spécifier leurs informations de compte. Si le compte Invité de la base de données est désactivé ou ne possède aucun jeu de privilèges d'accès activé comprenant un privilège d'accès étendu pour la Publication Web personnalisée, le moteur de publication Web utilise une authentification de base HTTP pour demander aux utilisateurs Web de s'authentifier. Le navigateur de l'utilisateur Web affiche la boîte de dialogue d'authentification de base HTTP pour permettre à l'utilisateur de saisir le nom d'utilisateur et le mot de passe d'un compte bénéficiant d'un privilège d'accès étendu pour la Publication Web personnalisée.

La liste suivante résume le processus qui se produit lorsqu'un utilisateur web utilise une solution de Publication Web personnalisée pour accéder à une base de données :

- Si vous n'avez pas défini de mot de passe pour un compte, les utilisateurs Web devront uniquement spécifier le nom de compte.
- Si le compte Invité est désactivé, les utilisateurs sont invités à donner un nom de compte et un mot de passe lorsqu'ils accèdent à une base de données. Un privilège d'accès étendu pour la Publication Web personnalisée doit être activé pour ce compte.
- Si le compte Invité est activé et bénéficie d'un jeu de privilèges comprenant un privilège d'accès étendu pour la Publication Web personnalisée, tous les utilisateurs Web peuvent automatiquement ouvrir la base de données avec les privilèges d'accès du compte Invité. Si le privilège étendu de Publication Web personnalisée est attribué au compte Invité :
  - Les utilisateurs Web n'ont pas besoin de spécifier de nom de compte ni de mot de passe lorsqu'ils ouvrent un fichier.
  - Tous les utilisateurs Web se connectent automatiquement avec le compte Invité et héritent des privilèges d'accès de celui-ci. Vous pouvez permettre aux utilisateurs de changer de compte de connexion via leur navigateur Web grâce à l'action de script Reconnexion (par exemple, pour basculer du compte Invité à un autre compte, jouissant de privilèges plus étendus).
  - Le jeu de privilèges par défaut des comptes Invité fournit un accès en « lecture seule ». Vous pouvez modifier les privilèges par défaut de ce compte, notamment les privilèges étendus. Consultez l'[Aide FileMaker Pro](#).
- Lorsqu'un utilisateur Web a saisi des informations de compte valides, celles-ci sont réutilisées tant que la session du navigateur n'a pas expiré. À partir du moment où la session du navigateur expire, l'utilisateur Web est de nouveau invité à saisir un compte valide.

**Remarque** Par défaut, les utilisateurs Web ne peuvent pas modifier le mot de passe de leur compte depuis un navigateur Web. Vous pouvez intégrer cette fonction dans une base de données grâce à l'action de script Modifier le mot de passe, qui permet aux utilisateurs Web de changer leur mot de passe depuis leur navigateur. Consultez l'[Aide FileMaker Pro](#).

## Protection de vos bases de données publiées

La fonction de Publication Web personnalisée vous permet de limiter l'accès à vos bases de données publiées.

- Affectez des mots de passe aux comptes de base de données utilisés pour la Publication Web personnalisée.
- Activez uniquement un privilège d'accès étendu pour la Publication Web personnalisée dans les jeux de privilèges des comptes que vous souhaitez autoriser à accéder à vos bases de données publiées.
- Désactivez le privilège d'accès étendu pour la Publication Web personnalisée pour une base de données spécifique en désélectionnant le privilège étendu fmxml ou fmphp pour tous les jeux de privilèges d'accès de cette base de données. Consultez l'[Aide FileMaker Pro](#).
- Activez ou désactivez la Publication Web personnalisée pour toutes les solutions de Publication Web personnalisée à l'aide de l'interface de ligne de commande (CLI). En outre, utilisez l'Admin Console de FileMaker Server pour démarrer ou arrêter le moteur de publication Web. Consultez l'[Aide FileMaker Server](#).
- Configurez votre serveur Web de façon à restreindre les adresses IP ayant accès à vos bases de données via le moteur de publication Web. Vous pouvez par exemple indiquer que seuls les utilisateurs Web utilisant l'adresse IP 192.168.100.101 sont autorisés à accéder à vos bases de données. Pour plus d'informations sur la restriction d'adresses IP, consultez la documentation de votre serveur Web.

FileMaker Server prend en charge le cryptage des données écrites sur disque et des données transférées aux clients.

- Cryptez votre base de données à l'aide de la fonction Cryptage de base de données de FileMaker Pro. Le cryptage protège le fichier de base de données FileMaker Pro et tous les fichiers temporaires écrits sur le disque. Consultez le [Guide d'installation et de configuration de FileMaker Server](#) et l'[Aide FileMaker Pro](#).
  - Une base de données cryptée hébergée sur FileMaker Server s'ouvre à l'aide de l'Admin Console ou de l'interface de ligne de commande (CLI). En tant qu'administrateur de FileMaker Server, ouvrez le fichier avec son mot de passe de cryptage de base de données afin que les clients FileMaker puissent utiliser la base de données cryptée.
  - Une fois que l'administrateur de FileMaker Server a ouvert la base de données cryptée FileMaker avec le mot de passe de cryptage, les clients FileMaker n'ont pas besoin du mot de passe de cryptage pour accéder à la base de données cryptée. Pour en savoir plus sur l'ouverture d'une base de données cryptée, consultez l'[Aide FileMaker Server](#).
- Utilisez le cryptage Secure Socket Layer (SSL) pour les communications entre le serveur Web et les navigateurs Web. Une connexion HTTPS permet d'accéder aux connexions SSL. FileMaker Server fournit un certificat SSL signé par Claris International Inc. qui ne vérifie pas le nom du serveur. Le certificat par défaut de FileMaker est uniquement destiné à des fins de tests. Un certificat SSL personnalisé est requis à des fins de production. Consultez le [Guide de démarrage de FileMaker Server](#).

Lorsque vous importez un certificat SSL personnalisé, les connexions de clients au serveur de base de données utilisent SSL et les connexions HTTP sont acheminées vers HTTPS. Avec un certificat SSL personnalisé, utilisez le répertoire HTTPS pour l'hébergement des fichiers PHP du site. Consultez la section chapitre 7, « Étapes générales de la Publication Web personnalisée avec PHP ».

Pour en savoir plus sur la sécurisation de vos bases de données, consultez le [Guide de la sécurité FileMaker](#).

**Remarque** Pour des raisons de sécurité, les pages Web hébergées par d'autres serveurs Web ne peuvent pas utiliser de balises <iframe> pour incorporer du contenu de Publication Web personnalisée. Pour incorporer du contenu de Publication Web personnalisée dans les balises <iframe> de pages Web distinctes, ces pages doivent être hébergées par le serveur Web de FileMaker Server.

## Prise en charge du serveur Web pour les types de média Internet (MIME)

Votre serveur Web détermine la prise en charge des types MIME (Multipurpose Internet Mail Extensions) actuellement enregistrés pour Internet. Le moteur de publication Web ne change pas la prise en charge MIME par le serveur Web. Consultez la documentation de votre serveur Web.

## À propos de la publication du contenu des rubriques Conteneur sur le Web

Le contenu d'une rubrique Conteneur peut être imbriqué dans la base de données, lié par référence à l'aide d'un chemin relatif ou stocké en externe.

### Objets de rubriques Conteneur imbriqués dans une base de données

Si une rubrique Conteneur stocke les fichiers réels dans la base de données FileMaker Pro, toute action sur le contenu de la rubrique Conteneur est inutile lorsque le fichier de base de données est correctement hébergé et facile d'accès sur FileMaker Server. Consultez la section « À propos de la syntaxe d'URL pour les objets FileMaker Pro de type Conteneur dans les solutions XML », page 27.

### Rubriques Conteneur avec des références de fichier stockées

Si une rubrique Conteneur contient une référence de fichier, vous devez procéder comme suit pour publier les fichiers référencés à l'aide du moteur de publication Web.

1. Stockez les fichiers d'objet Conteneur dans le dossier Web, situé dans le dossier FileMaker Pro.
2. Dans FileMaker Pro, insérez les objets dans la rubrique Conteneur et sélectionnez l'option **Stocker uniquement la référence au fichier**.
3. Copiez ou déplacez les fichiers d'objet référencé du dossier Web vers le même emplacement relatif, dans le dossier racine du logiciel de serveur.
  - Pour IIS (Windows) :  
`[disque]:\Program Files\FileMaker\FileMaker Server\HTTPServer\conf`  
où [disque] correspond au lecteur sur lequel réside le composant Moteur de publication Web de votre déploiement FileMaker Server.
  - Pour Apache (macOS) : `/Bibliothèque/FileMaker Server/HTTPServer/htdocs`



**Remarque** Pour les objets Conteneur stockés sous forme de référence externe, votre serveur Web doit être configuré pour prendre en charge les types MIME correspondant aux types de fichiers que vous souhaitez publier, par exemple des vidéos. Votre serveur Web détermine la prise en charge des types MIME actuellement enregistrés pour Internet. Le moteur de publication Web ne change pas la prise en charge MIME par le serveur Web. Consultez la documentation de votre serveur Web.

## Types de données de conteneur avec données stockées en externe

Si une rubrique Conteneur stocke des objets en externe (en d'autres termes, si vous avez sélectionné **Stocker les données de conteneur en externe** dans la boîte de dialogue Options pour rubrique FileMaker Pro), utilisez FileMaker Pro pour transférer des fichiers de base de données depuis le système de fichiers clients vers FileMaker Server. Lorsque vous utilisez FileMaker Pro pour télécharger une base de données, les données de la rubrique Conteneur stockées en externe sont téléchargées vers FileMaker Server dans le cadre du processus. Pour en savoir plus sur le transfert de fichiers de base de données vers FileMaker Server, consultez l'[Aide FileMaker Pro](#).

Lorsque vous téléchargez manuellement une base de données qui utilise une rubrique Conteneur avec des objets stockés en externe, vous devez exécuter ces étapes pour publier les objets Conteneur stockés en externe en utilisant le moteur de Publication Web.

### Pour télécharger manuellement une base de données :

1. Placez le fichier de base de données à l'emplacement adéquat sur le serveur. Placez les fichiers de base de données FileMaker Pro que FileMaker Server doit ouvrir (ou leurs raccourcis (Windows) ou alias (macOS)) dans les dossiers suivants :
  - Windows :  
[disque]:\Program Files\FileMaker\FileMaker Server\Data\Databases\  
où [disque] correspond au lecteur principal servant au démarrage de votre système.
  - macOS : /Bibliothèque/FileMaker Server/Data/Databases/Vous pouvez également placer les fichiers dans un dossier de bases de données supplémentaire défini en option.
2. Dans le dossier où vous avez placé la base de données, créez un dossier nommé RC\_Data\_FMS, s'il n'existe pas encore.
3. Dans le dossier RC\_Data\_FMS, créez un dossier dont le nom correspond à celui de votre base de données. Par exemple, si votre base de données s'appelle Clients, créez un dossier nommé Clients. Placez les objets stockés en externe dans le dossier que vous venez de créer.

**Remarque** Lorsque les bases de données sont hébergées sur FileMaker Server, plusieurs bases de données ne peuvent en aucun cas partager un dossier commun d'objets Conteneur. Les objets Conteneur de chacune des bases de données doivent se trouver dans un dossier identifié par ce nom de base de données.

4. Pour les fichiers partagés depuis macOS, changez les fichiers de façon à ce qu'ils appartiennent au groupe **fmsadmin**.

Pour en savoir plus sur le téléchargement manuel de bases de données, consultez l'[Aide FileMaker Server](#).

## Rubriques Conteneur et téléchargement progressif

Le moteur de Publication Web prend en charge un téléchargement progressif des fichiers audio (.mp3), des fichiers vidéo (.mov, .mp4 et .avi recommandés) et des fichiers PDF pour les conteneurs interactifs. Par exemple, un utilisateur Web peut commencer à visionner un film, même si la totalité du fichier vidéo n'est pas encore téléchargée. Pour rendre ce téléchargement progressif possible, vous devrez créer les fichiers en utilisant des options qui prennent en charge la lecture en direct ou qui sont optimisés pour s'afficher sur le Web. Par exemple, créez les fichiers PDF à l'aide de l'option d'optimisation en vue d'un affichage Web.

Lorsque vous importez un certificat SSL personnalisé, les connexions de clients au serveur de base de données utilisent SSL et les connexions HTTP sont acheminées vers HTTPS. FileMaker Server utilise des connexions sécurisées pour transmettre des données via HTTPS.

- Les données du conteneur interactif sont téléchargées via HTTPS.
- Les données sont tout aussi sécurisées que si la solution hébergée était une base de données locale, étant donné qu'aucun fichier du cache temporaire n'est créé et que les données sont cryptées pendant le transfert.

Lorsque vous ne disposez pas d'un certificat SSL personnalisé, les connexions que FileMaker Server utilise pour transférer des données ne sont pas cryptées pendant le transfert et les données sont transférées via HTTP.

- Les clients FileMaker voient les données du conteneur interactif avec un retard minime.
- FileMaker Server met en cache les données de la rubrique Conteneur dans un dossier cache sur le serveur lorsqu'un client FileMaker Pro, Claris FileMaker Go® ou Web interroge les données. Les données peuvent rester dans le dossier cache du serveur pendant deux heures, jusqu'à ce que FileMaker Server le vide périodiquement. Les données ne sont pas mises en cache localement sur le client.

## Affichage des données d'une rubrique Conteneur par les utilisateurs Web

Lorsque vous publiez une base de données à l'aide d'un moteur de publication Web, les limites suivantes s'appliquent aux données des rubriques Conteneur :

- Les utilisateurs Web ne peuvent pas modifier ni compléter le contenu des rubriques Conteneur. Les utilisateurs Web ne peuvent pas utiliser les types de données de conteneur pour télécharger des objets vers la base de données.
- Pour les bases de données utilisant une rubrique Conteneur dans laquelle les miniatures sont activées, le moteur de publication Web télécharge l'intégralité du fichier et pas uniquement une miniature.

## Scripts FileMaker et Publication Web personnalisée

La fonction Gérer les scripts de FileMaker Pro peut automatiser les tâches fréquentes et combiner plusieurs tâches. Lorsque les scripts FileMaker sont utilisés avec la Publication Web personnalisée, ils permettent aux utilisateurs Web d'exécuter plusieurs tâches ou une série de tâches.

La plateforme FileMaker prend en charge de nombreuses actions de script dans la Publication Web personnalisée. Les utilisateurs Web peuvent effectuer différentes tâches automatisées lorsque vous utilisez des scripts dans une chaîne de requête pour une URL. Afin de voir les actions de script que prend en charge la Publication Web personnalisée, cliquez sur le bouton **Compatibilité** dans la fenêtre Espace de travail de script de FileMaker Pro, puis sélectionnez **Publication Web personnalisée**. Les actions de script qui n'apparaissent pas en grisé sont prises en charge pour la Publication Web personnalisée. Pour en savoir plus sur la création de scripts, consultez l'[Aide FileMaker Pro](#).

## Astuces et considérations à propos des scripts

Bien qu'un grand nombre d'actions de script fonctionnent de manière identique sur le Web, plusieurs fonctionnent différemment. Consultez la section « Comportement des scripts dans les solutions de Publication Web personnalisée », page 20. Avant de partager votre base de données, testez tous les scripts qui seront exécutés depuis un navigateur Web. Veillez à vous connecter avec différents comptes utilisateur, afin de vous assurer qu'ils fonctionnent comme prévu pour tous les clients. Consultez le fichier journal du moteur de publication Web (wpe.log) pour rechercher les éventuelles erreurs liées aux scripts. Consultez la section « Utilisation du journal du moteur de publication Web », page 96.

Gardez à l'esprit les astuces et considérations suivantes :

- Pensez aux valeurs qu'un script doit renvoyer. Préparez-vous à gérer toutes les données renvoyées. Dans FileMaker Pro, un script peut renvoyer tous les enregistrements d'une table ou d'un jeu d'enregistrements trouvé. Toutefois, si un script renvoie tous les enregistrements d'une table, une application Web qui tente de tous les traiter peut rencontrer un problème de mémoire. Pensez à utiliser le paramètre de requête `-max` avec les requêtes XML ou la méthode `setRange()` avec les requêtes PHP pour limiter le nombre d'enregistrements renvoyés.
- Utilisez les comptes et les privilèges pour restreindre l'ensemble des scripts pouvant être exécutés par un utilisateur Web. Vérifiez que les scripts contiennent exclusivement des actions compatibles avec le Web et donnent uniquement accès aux scripts pouvant être utilisés depuis un navigateur Web.
- Pensez aux effets secondaires des scripts qui exécutent une combinaison d'actions contrôlées par des privilèges. Par exemple, si un script comporte une action visant à supprimer des enregistrements et qu'un utilisateur Web ne se connecte pas avec un compte permettant la suppression d'enregistrements, le script n'exécutera pas l'action Supprimer des enregistrements. L'exécution du script peut néanmoins se poursuivre et mener à des résultats inattendus.
- Dans la fenêtre Espace de travail de script, accordez les privilèges d'accès intégral à un script afin d'effectuer des actions auxquelles les utilisateurs individuels n'ont normalement pas accès. Par exemple, vous pouvez empêcher les utilisateurs de supprimer des enregistrements avec leurs comptes et leurs privilèges, tout en leur permettant d'exécuter un script visant à supprimer certains types d'enregistrements dans les conditions définies au sein du script.
- Pour autoriser les scripts à installer des plug-ins pour les solutions de Publication Web personnalisée et FileMaker WebDirect, utilisez l'Admin Console de FileMaker Server pour activer les paramètres nécessaires. Dans **Plug-ins de publication Web**, définissez les paramètres **Plug-ins de publication Web** et **Action de script Installer le fichier de plug-in** sur **Activé**. Afin d'éviter qu'un script n'installe de plug-ins pour les solutions de publication Web personnalisée, définissez-les sur **Désactivé**.
- Certains scripts fonctionnant avec une action d'un client FileMaker Pro peuvent nécessiter une action de script Valider enreg./requêtes supplémentaire pour permettre l'enregistrement des données sur l'hôte. Comme les utilisateurs Web ne bénéficient pas d'une connexion directe à l'hôte, ils ne sont pas informés des modifications de données. Par exemple, des fonctions telles que les listes de valeurs conditionnelles ne présentent pas la même réactivité pour les utilisateurs Web parce que les données doivent être enregistrées sur l'hôte avant que leurs effets soient visibles dans la rubrique Liste de valeurs.

- Tout script modifiant des données doit inclure l'action de script Valider enreg/requêtes, car toutes les modifications de données ne sont pas visibles dans le navigateur tant que les données ne sont pas sauvegardées ou « soumises » au serveur. Ceci inclut plusieurs actions de script telles que Couper, Copier ou Coller. De nombreuses actions de script uniques doivent être converties en script pour inclure l'action Valider enreg./requêtes. Lors de la conception de scripts destinés à être exécutés depuis un navigateur Web, incorporez l'action Valider enreg./requêtes en fin de script pour garantir l'enregistrement de toutes les modifications.
- Il se peut que les scripts dépendant de données qui ont été modifiées mais pas encore validées ne fonctionnent pas comme prévu avec un client FileMaker Pro. Par exemple, si une rubrique contient un calcul Obtenir(RubriquesModifiées), lorsque l'enregistrement est modifié via une requête XML ou un appel PHP, le calcul Obtenir(RubriquesModifiées) renvoie une liste vide. Cependant, si la solution de Publication Web personnalisée exécute un script FileMaker qui modifie l'enregistrement, le calcul Obtenir(RubriquesModifiées) renvoie une liste de rubriques modifiées par le script.
- Pour créer des scripts conditionnels basés sur le type de client, utilisez la fonction Obtenir(VersionApplication). Si la valeur renvoyée comprend une chaîne « moteur de publication Web », vous savez que l'utilisateur actuel accède à votre base de données à l'aide de la Publication Web personnalisée. Pour en savoir plus sur les fonctions, consultez l'[Aide FileMaker Pro](#).
- Ouvrez chaque script susceptible d'être exécuté par les utilisateurs Web et vérifiez que le script s'exécute correctement lorsque la base de données est hébergée en tant que solution de Publication Web personnalisée. Vérifiez que le script utilise uniquement des actions de script étant prises en charge pour la Publication Web personnalisée, tel que décrit ci-dessus.

## Comportement des scripts dans les solutions de Publication Web personnalisée

Le comportement de certaines actions de script utilisées dans les solutions de Publication Web personnalisée diffère par rapport à celui de FileMaker Pro. Pour en savoir plus sur la compatibilité, consultez l'[Aide FileMaker Pro](#).

Les scripts de solutions de Publication Web personnalisée ne peuvent pas exécuter de script dans d'autres fichiers FileMaker Pro, à moins que ces fichiers ne soient hébergés sur la même installation de FileMaker Server et que le même privilège étendu de Publication Web personnalisée ne soit activé sur ces autres fichiers.

## Déclencheurs de scripts et solutions de Publication Web personnalisée

Dans FileMaker Pro, les déclencheurs de scripts peuvent être activés à la fois par des actions de script et des actions de l'utilisateur (par exemple en cliquant sur une rubrique). Cependant, dans la Publication Web personnalisée, les scripts sont les seuls à pouvoir activer les déclencheurs de script.

Pour plus d'informations sur les déclencheurs de script, consultez l'[Aide FileMaker Pro](#).

**Remarque** Dans les solutions de Publication Web personnalisée, le déclencheur de script SurOuverturePremiereFenetre n'est pas activé. Le déclencheur de script SurFermetureDerniereFenetre est activé uniquement lorsqu'un script ferme la dernière fenêtre virtuelle. Vous pouvez exécuter les scripts manuellement à l'aide du paramètre de requête -script XML ou de la méthode PHP `newPerformScriptCommand()`.

# Chapitre 3

## À propos de la fonction de Publication Web personnalisée utilisant le langage XML

### Création de sites Web dynamiques avec le moteur de publication Web

Le moteur de publication Web dote FileMaker Server de la Publication Web personnalisée en utilisant la publication de données XML. La Publication Web personnalisée offre divers avantages :

- **Personnalisation** : Vous pouvez déterminer comment les utilisateurs Web interagissent avec les données FileMaker et comment les données s'affichent dans les navigateurs Web.
- **Echange de données** : L'utilisation du format XML de FileMaker vous permet d'échanger des données FileMaker avec d'autres sites Web et d'autres applications.
- **Intégration des données** : Vous pouvez intégrer des données FileMaker dans d'autres sites Web, avec d'autres middlewares et avec des applications personnalisées. Vous pouvez faire en sorte que les données s'affichent comme si elles provenaient d'un autre site Web au lieu d'afficher un modèle FileMaker Pro complet dans le navigateur Web.
- **Sécurité** : L'administrateur FileMaker Server peut activer ou désactiver les fonctions de Publication Web XML pour toutes les bases de données hébergées sur le serveur. En tant que propriétaire de la base de données FileMaker Pro, vous pouvez contrôler l'accès des utilisateurs Web à la Publication Web XML pour chaque base de données.
- **Contrôle et filtrage des données publiées** : Vous pouvez contrôler et filtrer les données et le type des données contenues dans la base de données que vous souhaitez publier tout en empêchant l'utilisation non autorisée de la base de données. Vous pouvez également masquer les métadonnées, par exemple les noms de la base de données et de la rubrique.
- **Utilisation d'une norme ouverte** : Les outils, les ressources et le personnel compétent auxquels vous avez accès pour les solutions de Publication Web personnalisée sont plus importants. Si vous connaissez déjà le langage XML standard, vous pouvez vous lancer immédiatement dans le développement de solutions. Vous devez néanmoins connaître la syntaxe d'URL et les paramètres de requête à employer pour utiliser la Publication Web personnalisée avec XML.

La Publication Web personnalisée avec XML vous permet de récupérer des données dans les bases de données FileMaker Pro et de les utiliser facilement dans d'autres formats de sortie. L'utilisation d'une requête HTTP avec les commandes et les paramètres appropriés vous aide à récupérer les données FileMaker sous la forme d'un document XML. Vous pouvez alors exporter les données XML vers d'autres applications. Consultez la section « Accès aux données XML via le moteur de publication Web », page 29.

## Fonctions clés de la Publication Web personnalisée avec XML

La Publication Web personnalisée FileMaker Server avec XML fournit plusieurs nouvelles fonctionnalités importantes :

- Les bases de données sont hébergées sur FileMaker Server et l'application FileMaker Pro ne doit pas forcément être lancée.
- Vous pouvez utiliser le traitement JavaScript côté serveur de XML.
- Comme dans FileMaker Pro, l'accès aux données, aux modèles et aux rubriques est basé sur les paramètres de compte d'utilisateur définis dans les privilèges d'accès de la base de données. Le moteur de publication Web présente également plusieurs autres améliorations en matière de sécurité. Consultez la section « Protection de vos bases de données publiées », page 15.
- Les utilisateurs Web peuvent exécuter des scripts complexes contenant plusieurs actions. La plateforme FileMaker prend en charge de nombreuses actions de script dans la Publication Web personnalisée. Consultez la section « Scripts FileMaker et Publication Web personnalisée », page 18.
- Vous pouvez transmettre une valeur de paramètre à un script FileMaker. Consultez les sections « Paramètre de requête –script.param (transfert du paramètre au script) », page 60, « Paramètre de requête –script.prefind.param (transfert du paramètre au script avant recherche) », page 61 et « Paramètre de requête –script.presort.param (transfert du paramètre au script avant recherche) », page 61.
- La grammaire XML `fmresultset` vous permet d'accéder aux rubriques par leur nom et de manipuler les données `relatedset` (table externe).
- Pour accéder aux données d'une base de données, vous devez indiquer un modèle. Consultez la section chapitre 5, « Noms corrects utilisés dans les chaînes de requête XML ».

## Configuration pour la publication Web

### Configuration requise pour la publication d'une base de données à l'aide de la Publication Web personnalisée

Pour publier des bases de données à l'aide de la fonction de Publication Web personnalisée avec XML, il vous faut :

- un déploiement FileMaker Server comportant :
  - un serveur Web, soit Microsoft IIS (Windows), soit Apache (macOS) ;
  - le serveur de bases de données FileMaker, avec option Publication Web personnalisée activée à l'aide de l'interface de ligne de commande (CLI) ;
  - le moteur de publication Web installé et configuré ;
- une ou plusieurs bases de données FileMaker Pro hébergées par FileMaker Server ;
- l'adresse IP ou le nom de domaine de l'hôte exécutant le serveur Web ;
- un navigateur Web et un accès au serveur Web pour développer et tester la solution de publication Web personnalisée.

Consultez le [d'installation et de configuration de FileMaker Server](#).

## Éléments devant être employés par les utilisateurs Web pour accéder à une solution de publication Web personnalisée

Pour accéder à une solution de Publication Web personnalisée utilisant XML, les utilisateurs Web doivent posséder :

- un navigateur Web ;
- un accès à Internet ou à un intranet, ainsi qu'un accès au serveur Web ;
- l'adresse IP ou le nom de domaine de l'hôte exécutant le serveur Web.

Si la base de données est protégée par un mot de passe, les utilisateurs Web doivent également saisir un nom d'utilisateur et un mot de passe pour accéder au compte de la base de données.

## Connexion à Internet ou à un intranet

Lorsque vous publiez des bases de données sur Internet ou sur un intranet, l'ordinateur hôte doit exécuter FileMaker Server et les bases de données que vous souhaitez partager doivent être hébergées et accessibles. En outre :

- Publiez votre base de données sur un ordinateur possédant une connexion permanente à Internet ou à un intranet. En effet, si la connexion n'est pas permanente, les utilisateurs Web doivent attendre que l'hôte se connecte à Internet ou à un intranet pour accéder à la base de données.
- L'ordinateur hôte du serveur Web, partie intégrante du déploiement FileMaker Server, doit posséder une adresse IP *statique* (permanente) dédiée ou un nom de domaine. Si vous vous connectez à Internet avec un fournisseur d'accès à Internet (FAI), votre adresse IP est peut-être *allouée de manière dynamique* (c'est-à-dire différente à chaque fois que vous vous connectez). Le cas échéant, les utilisateurs Web auront plus de difficultés à localiser une base de données. Si vous ne savez pas de quel type d'accès vous disposez, contactez votre fournisseur d'accès à Internet ou l'administrateur du réseau.

## Étape suivante

Voici quelques suggestions pour commencer à développer des solutions de publication Web personnalisée :

- Si ce n'est pas déjà fait, utilisez l'interface de ligne de commande (CLI) pour activer la Publication Web personnalisée. Consultez l'[Aide FileMaker Server](#).
- Dans FileMaker Pro, ouvrez chaque base de données FileMaker Pro à publier et assurez-vous que chacune d'entre elles dispose des privilèges d'accès étendus pour la Publication Web personnalisée. Consultez la section « Activation de la Publication Web personnalisée dans une base de données », page 13.
- Pour savoir comment accéder aux données dans les bases de données FileMaker Pro en utilisant le langage XML, consultez la section « Accès aux données XML via le moteur de publication Web », page 29.

# Chapitre 4

## Accès aux données XML avec le moteur de publication Web

Vous pouvez utiliser et mettre à jour des données FileMaker au format XML (Extensible Markup Language) à l'aide du moteur de publication Web. Un grand nombre d'utilisateurs, d'organisations et d'entreprises utilisent le langage XML pour transférer des informations sur les produits, des transactions, des informations sur le stock, ainsi que d'autres données commerciales.

### Utilisation de la Publication Web personnalisée avec XML

Si vous connaissez déjà le langage XML standard, vous pouvez immédiatement à utiliser le moteur de publication Web. Vous devez néanmoins apprendre la syntaxe d'URL et connaître les paramètres de requête à employer pour la Publication Web personnalisée avec XML.

Grâce aux requêtes URL HTTP avec les commandes et les paramètres de requête de FileMaker, vous pouvez interroger une base de données hébergée par FileMaker Server et télécharger les données obtenues au format XML. Par exemple, vous pouvez interroger une base de données pour obtenir tous les enregistrements relatifs à un code postal donné, puis employer les données XML obtenues comme bon vous semble.

Consultez la [Base de connaissances](#).

**Remarque** Le moteur de Publication Web génère des données XML correctement formatées et conformes à la spécification XML 1.0. Pour plus de détails sur les conditions requises pour le formatage des données XML, reportez-vous aux spécifications XML disponibles à l'adresse [www.w3.org](http://www.w3.org).

### Comparaison du moteur de publication Web avec l'importation et l'exportation XML

Le moteur de Publication Web et FileMaker Pro permettent tous les deux d'utiliser des données XML avec des bases de données FileMaker Pro. Cependant, il existe des différences importantes entre ces deux méthodes :

- Pour accéder aux données XML, le moteur de publication Web prend en charge les grammaires `fmresultset`, `FMPXMLRESULT` et `FMPXMLLAYOUT`. Pour l'importation XML, FileMaker Pro utilise la grammaire `FMPXMLRESULT`, et pour l'exportation, il utilise la grammaire `FMPXMLRESULT`. Consultez la section « Accès aux données XML via le moteur de publication Web », page 29.
- Pour accéder aux données XML à l'aide du moteur de publication Web, utilisez une chaîne de requête du moteur de publication Web dans une URL. Pour importer et exporter des données XML avec FileMaker Pro, utilisez les commandes ou les scripts du menu FileMaker Pro.
- Le moteur de publication Web est basé sur le serveur et peut être installé sur le même hôte que FileMaker Server ou sur un hôte différent. L'importation et l'exportation de données XML de FileMaker Pro se font à partir d'un ordinateur de bureau.
- Vous pouvez accéder de façon dynamique aux données XML à partir des bases de données FileMaker Pro en utilisant des requêtes URL avec le moteur de publication Web. La fonction d'exportation de données XML de FileMaker Pro génère un fichier de données XML spécifié à l'avance.

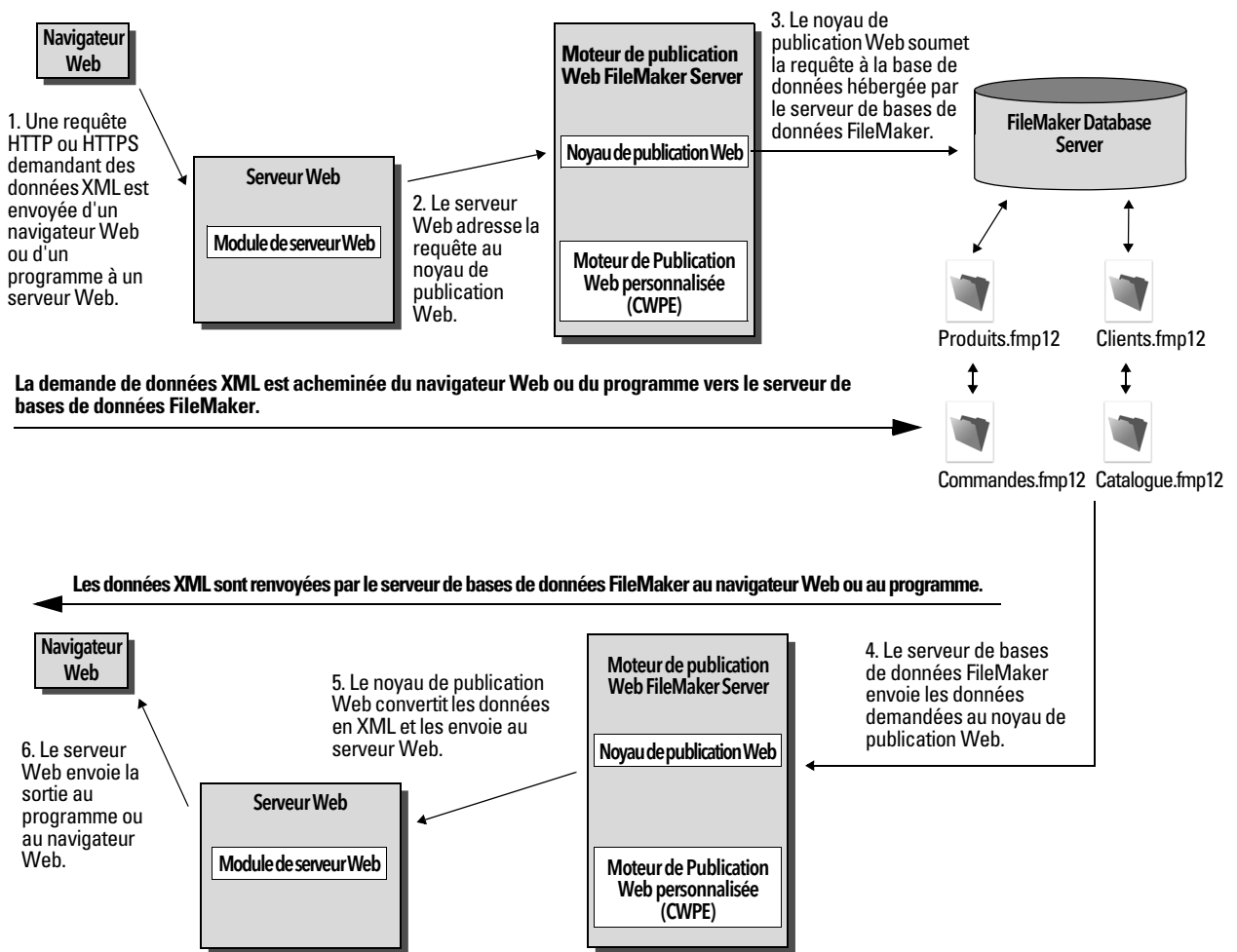


- Les opérations sur les données XML via un moteur de publication Web sont interactives. L'importation et l'exportation de données XML de FileMaker Pro se font selon un processus par lot.
- Contrairement à FileMaker Pro, le moteur de publication Web peut accéder aux données XML à partir d'une table externe FileMaker Pro.
- Contrairement à FileMaker Pro, le moteur de publication Web peut accéder aux données d'une rubrique de type Conteneur.
- Contrairement à FileMaker Pro, le moteur de publication Web fournit un accès en temps réel aux données FileMaker via HTTP ou HTTPS.

**Remarque** Pour en savoir plus sur l'utilisation de FileMaker Pro et effectuer des importations et des exportations de données au format XML, consultez l'[Aide FileMaker Pro](#).

### Génération de données XML par le moteur de publication Web à partir d'une requête

Une fois qu'une demande de données XML a été envoyée au serveur Web, le moteur de publication Web interroge la base de données FileMaker Pro et renvoie les données sous forme de document XML.



## Processus général d'accès aux données XML à partir du moteur de publication Web

Voici un aperçu du processus consistant à employer le moteur de publication Web pour accéder aux données XML dans une base de données FileMaker Pro.

1. Utilisez l'interface de ligne de commande (CLI) pour activer la publication XML. Consultez l'[Aide FileMaker Server](#).

2. Dans FileMaker Pro, ouvrez chaque base de données FileMaker Pro à publier et assurez-vous que pour chacune d'entre elles, le privilège étendu fmxml est activé pour la fonction de Publication Web personnalisée utilisant XML. Consultez la section « Activation de la Publication Web personnalisée dans une base de données », page 13.

Pour accéder aux données XML dans une table externe, définissez l'affichage de modèle de base de données sur **Afficher sous forme de formulaire** ou sur **Afficher sous forme de liste**. Si un utilisateur ou un script modifie l'affichage du modèle de base de données et le définit sur **Afficher sous forme de tableau**, seul le premier enregistrement (la première rangée de la table externe) est accessible sous forme de données XML.

Les données XML sortent dans un ordre qui correspond à l'ordre dans lequel les objets de la rubrique ont été ajoutés au modèle. Si vous souhaitez que l'ordre des données XML corresponde à l'ordre dans lequel les rubriques apparaissent à l'écran (du haut vers le bas et de la gauche vers la droite), sélectionnez toutes les rubriques, regroupez-les, puis dégroupuez-les. Cette procédure réinitialise l'ordre du modèle pour le faire correspondre à l'ordre de l'écran.

3. Dans l'Admin Console de FileMaker Server, confirmez que le privilège étendu fmxml est activé pour chaque base de données FileMaker Pro publiée.

- Dans l'Admin Console, cliquez sur la page **Bases de données**.
- Sélectionnez **Afficher les privilèges étendus** dans le sélecteur de menu à côté de **Toutes les bases de données**. Le texte **FMXML** s'affiche pour les bases de données dont le privilège étendu fmxml est activé.

4. Envoyez une requête HTTP ou HTTPS sous la forme d'une URL qui spécifie la grammaire XML FileMaker, une commande de requête, puis un ou plusieurs paramètres de requête FileMaker au moteur de publication Web, par l'intermédiaire d'un formulaire HTML, d'un lien HREF ou d'un script dans votre programme ou votre page Web. Vous pouvez également entrer l'URL dans un navigateur Web.

Pour plus d'informations sur la spécification d'une URL, consultez la section « À propos de la syntaxe d'URL pour les données XML et les objets de type Conteneur ». Pour plus d'informations sur les commandes de requête et sur les paramètres, consultez les sections « Utilisation de chaînes de requête FileMaker pour rechercher des données XML », page 39, et chapitre 5, « Noms corrects utilisés dans les chaînes de requête XML ».

5. Le moteur de publication Web utilise la grammaire spécifiée dans l'URL et génère les données XML contenant les résultats de votre requête, par exemple un jeu d'enregistrements de la base de données, puis les renvoie à votre programme ou à votre navigateur Web.

6. S'il est équipé d'un analyseur XML, le navigateur Web affiche les données, ou le programme utilise les données de la façon que vous avez indiquée.

## À propos de la syntaxe d'URL pour les données XML et les objets de type Conteneur

Cette section décrit la syntaxe d'URL à utiliser au niveau du moteur de publication Web pour accéder aux données XML et aux objets de type Conteneur à partir de bases de données FileMaker Pro.

### À propos de la syntaxe d'URL pour les données XML

La syntaxe d'URL à employer pour utiliser le moteur de publication Web et accéder aux données XML à partir de bases de données FileMaker Pro est la suivante :

```
<scheme>://<host>[:<port>]/fmi/xml/<xml_grammar>.xml[?<query string>]
```

où :

- `<scheme>` peut correspondre au protocole HTTP ou HTTPS.
- `<host>` correspond à l'adresse IP ou au nom de domaine de l'hôte sur lequel le serveur Web est installé.
- `<port>` est facultatif et spécifie le port utilisé par le serveur Web. Si aucun port n'est précisé, le port par défaut du protocole est utilisé (port 80 pour HTTP ou port 443 pour HTTPS).
- `<xml_grammar>` est le nom de la grammaire XML FileMaker. Les valeurs possibles sont `fmresultset`, `FMPXMLRESULT` ou `FMPXMLLAYOUT`. Consultez les sections « Utilisation de la grammaire `fmresultset` », page 30 et « Utilisation d'autres grammaires XML FileMaker », page 34.
- `<query string>` associe une commande de requête à un ou plusieurs paramètres de requête pour la publication XML FileMaker. (la commande `-dbnames` ne requiert aucun paramètre). Consultez les sections « Utilisation de chaînes de requête FileMaker pour rechercher des données XML », page 39, et chapitre 5, « Noms corrects utilisés dans les chaînes de requête XML ».

**Remarque** La syntaxe d'URL, notamment les noms de commande et de paramètre de requête, applique la distinction majuscules-minuscules, à l'exception des parties de la chaîne de requête. La majorité des URL sont écrites en minuscules, à l'exception des deux noms de grammaire, qui sont en majuscules : `FMPXMLRESULT` et `FMPXMLLAYOUT`. Pour plus d'informations sur les règles de distinction majuscules-minuscules applicables à la chaîne de requête, consultez la section « Consignes d'utilisation des commandes et des paramètres de requête », page 44.

### Exemples

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=products&-lay=sales
&-findall
http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?-db=products&-lay=sales
&-findall
```

### À propos de la syntaxe d'URL pour les objets FileMaker Pro de type Conteneur dans les solutions XML

Dans un document XML généré pour une solution XML, la syntaxe à employer pour faire référence à un objet de type Conteneur est différente pour les rubriques Conteneur qui stockent l'objet concerné dans la base de données de celle pour les rubriques Conteneur qui stockent une référence à l'objet.

### Si une rubrique Conteneur stocke l'objet lui-même dans la base de données

L'élément `<data>` de la rubrique Conteneur utilise la syntaxe d'URL suivante pour faire référence à l'objet :

```
<data>/fmi/xml/cnt/data.<extension>?<query string></data>
```

où `<extension>` est l'extension du nom de fichier qui identifie le type d'objet, par exemple `.jpg`. L'extension du nom de fichier définit le type MIME qui permet au navigateur Web d'identifier correctement les données de type Conteneur. Pour plus d'informations concernant `<query string>`, consultez la section précédente « À propos de la syntaxe d'URL pour les données XML ».

#### Exemple

```
<data>/fmi/xml/cnt/data.jpg?-db=products&-lay=sales  
&-field=product_image(1)&-recid=2</data>
```

**Remarque** Dans le document XML généré pour une rubrique Conteneur, la valeur du paramètre de requête `-field` est un nom de rubrique qualifié complet. Le nombre entre parenthèses indique le nombre de répétitions pour la rubrique Conteneur. Il est généré à la fois pour les rubriques multivaluées et pour les rubriques non multivaluées. Consultez la section « À propos de la syntaxe d'un nom de rubrique entièrement qualifié », page 46.

Pour extraire les données de conteneur de la base de données, utilisez la syntaxe suivante :

```
<scheme>://<host>[:<port>]/fmi/xml/cnt/data.<extension>?<query string>
```

Pour plus d'informations concernant `<scheme>`, `<host>` ou `<port>`, consultez la section « À propos de la syntaxe d'URL pour les données XML ».

#### Exemple

```
http://www.company.com/fmi/xml/cnt/data.jpg?-db=products&-lay=sales  
&-field=product_image(1)&-recid=2
```

### Si une rubrique Conteneur contient une référence à un fichier et non à un objet réel

L'élément `<data>` de la rubrique Conteneur contient un chemin d'accès relatif faisant référence à l'objet.

#### Exemple

```
<data>/images/logo.jpg</data>
```

**Remarque** L'objet Conteneur référencé doit être stocké dans le dossier Web de FileMaker Pro lorsque l'enregistrement est créé ou modifié, puis doit être copié ou déplacé dans un dossier possédant le même emplacement relatif dans le dossier racine du logiciel de serveur Web. Consultez la section « À propos de la publication du contenu des rubriques Conteneur sur le Web », page 16.

### Si une rubrique Conteneur est vide

L'élément `<data>` de la rubrique est vide.

## À propos du codage de texte URL

Les URL utilisées pour accéder aux données XML et aux objets de type Conteneur doivent absolument être codées au format UTF-8 (Unicode Transformation Format 8 bits). Consultez la section « À propos des données UTF-8 », page 39.

### Exemple

Pour définir la valeur de la rubrique « info » sur fiancée, vous pouvez utiliser l'URL suivante :

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=members
&-lay=relationships&-recid=2&info=fianc%C3%A9e&-edit
```

Dans cette URL, %C3%A9 est la représentation codée au format UTF-8 du caractère é.

Consultez la spécification URL sur le site [www.w3.org](http://www.w3.org).

## Accès aux données XML via le moteur de publication Web

Pour accéder aux données XML via le moteur de publication Web, utilisez une URL qui spécifie le nom de la grammaire FileMaker à employer, une commande de requête FileMaker, puis un ou plusieurs paramètres de requête FileMaker. A partir de votre base de données, le moteur de publication Web génère des données XML formatées à l'aide de l'un des types de grammaire suivants :

- **fmresultset** : Il s'agit de la grammaire que nous recommandons d'utiliser pour le moteur de publication Web lors de l'accès aux données XML. Elle est flexible et particulièrement adaptée à un accès par le nom simplifié aux rubriques et à une manipulation plus facile des données *relatedset* (table externe). Cette grammaire est également plus étroitement liée à la terminologie et aux fonctionnalités de FileMaker, notamment aux options de stockage de type Global et à l'identification des rubriques de type Statistique et Calcul. Pour faciliter la publication Web, cette grammaire se veut plus prolixue que la grammaire *FMPXMLRESULT*. Consultez la section « Utilisation de la grammaire fmresultset », page 30.
- **FMPXMLRESULT et FMPXMLLAYOUT** : Vous pouvez employer les grammaires *FMPXMLRESULT* et *FMPXMLLAYOUT* avec le moteur de publication Web pour accéder aux données XML. Pour utiliser une feuille de style dédiée à l'exportation XML et à la Publication Web personnalisée, utilisez la grammaire *FMPXMLRESULT*. Pour accéder aux listes de valeurs et aux informations d'affichage des rubriques dans des modèles, utilisez la grammaire *FMPXMLLAYOUT*. Consultez la section « Utilisation d'autres grammaires XML FileMaker », page 34.

En fonction de la grammaire spécifiée dans la requête de type URL, le moteur de publication Web génère un document XML utilisant l'une des grammaires disponibles. Chaque document XML comporte une déclaration d'espace de nom XML par défaut pour la grammaire. Consultez la section suivante « À propos des espaces de nom pour le langage XML de FileMaker ». Pour afficher et manipuler des données FileMaker au format XML, utilisez l'une de ces grammaires dans votre document ou dans votre page Web.

**Remarque** Les données XML générées par le moteur de publication Web sont codées au format UTF-8 (Unicode Transformation Format 8). Consultez la section « À propos des données UTF-8 », page 39.

## À propos des espaces de nom pour le langage XML de FileMaker

Des espaces de nom XML uniques aident à distinguer les balises XML de l'application à laquelle ils sont destinés. Par exemple, si vos documents XML contiennent deux éléments `<DATABASE>` destinés respectivement à des données XML FileMaker et à des données XML Oracle, les espaces de nom permettent d'identifier les éléments `<DATABASE>` pour chacun de ces types de données. Le moteur de publication Web génère un espace de nom par défaut pour chaque grammaire.

Pour cette grammaire	Cet espace de nom par défaut est généré
fmresultset	xmlns="http://www.filemaker.com/xml/fmresultset"
FMPXMLRESULT	xmlns="http://www.filemaker.com/fmpxmlresult"
FMPXMLLAYOUT	xmlns="http://www.filemaker.com/fmpxmllayout"

## À propos des codes d'erreurs des bases de données FileMaker Pro

Accessoirement, le moteur de publication Web renvoie un code d'erreur au début de chaque document XML répertoriant une erreur dans l'exécution de la commande de requête la plus récemment exécutée. La valeur zéro (0) indique qu'il n'y a pas d'erreur.

Pour cette grammaire	La syntaxe utilisée est
fmresultset	<error code="0"></error>
FMPXMLRESULT	<ERRORCODE>0</ERRORCODE>
FMPXMLLAYOUT	<ERRORCODE>0</ERRORCODE>

L'élément de code d'erreur dans le document XML indique des erreurs relatives à la base de données et aux chaînes de requête. Consultez la section annexe A, « Codes d'erreur de la Publication Web personnalisée ».

## Récupération des définitions de type de document pour les grammaires FileMaker

Utilisez une requête HTTP pour récupérer les définitions de type de document (DTD) pour les grammaires FileMaker.

Pour cette grammaire	Utilisez la requête HTTP suivante
fmresultset	http://<host>[:<port>]/fmi/xml/fmresultset.dtd
FMPXMLRESULT	http://<host>[:<port>]/fmi/xml/FMPXMLRESULT.dtd
FMPXMLLAYOUT	http://<host>[:<port>]/fmi/xml/FMPXMLLAYOUT.dtd

## Utilisation de la grammaire fmresultset

Les noms des éléments XML de cette grammaire utilisent la terminologie FileMaker et le stockage des rubriques est séparé des types de rubrique. La grammaire offre également la possibilité d'identifier les rubriques de type Statistique, Calcul et Global.

Pour utiliser la grammaire `fmresultset`, indiquez le nom de la grammaire `fmresultset` dans l'URL qui recherche le document XML à partir du moteur de publication Web :

```
fmresultset.xml
```

## Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-findall
```

**Remarque** Veillez à utiliser des minuscules pour spécifier la grammaire `fmresultset`.

Le moteur de publication Web génère alors un document XML utilisant la grammaire `fmresultset`. Dans le document XML, le moteur de Publication Web fera référence à la définition du type de document pour la grammaire `fmresultset` dans l'instruction `<!DOCTYPE>` située sur la deuxième ligne du document, immédiatement après l'instruction `<?xml...?>`. L'instruction `<!DOCTYPE>` indique l'URL à employer pour télécharger la DTD relative à la grammaire `fmresultset`.

## Description des éléments de la grammaire `fmresultset`

La grammaire `fmresultset` se compose principalement des éléments `<datasource>`, `<metadata>` et `<resultset>`.

### élément `<datasource>`

Dans la grammaire `fmresultset`, l'élément `<datasource>` contient les attributs `table`, `layout`, `date-format`, `time-format`, `timestamp-format`, `total-count` et `database`.

- L'attribut `date-format` de l'élément `<datasource>` indique le format des dates dans le document XML :

```
dd/MM/yyyy
```

où :

- `MM` est la valeur à 2 chiffres relative au mois (de 01 à 12, 01 correspondant au mois de janvier et 12, au mois de décembre)
- `dd` est la valeur à 2 chiffres relative au jour du mois (de 01 à 31)
- `yyyy` est la valeur à 4 chiffres relative à l'année
- L'attribut `time-format` de l'élément `<datasource>` indique le format des heures dans le document XML :

```
HH:mm:ss
```

où :

- `HH` est la valeur à 2 chiffres pour les heures (de 00 à 23 pour le format 24 heures)
- `mm` est la valeur à 2 chiffres pour les minutes (de 00 à 59)
- `ss` est la valeur à 2 chiffres pour les secondes (de 00 à 59)
- L'attribut `timestamp-format` de l'élément `<datasource>` associe les formats de date et d'heure dans un horodatage :

```
dd/MM/yyyy HH:mm:ss
```

### élément <metadata>

L'élément <metadata> de la grammaire `fmresultset` contient un ou plusieurs éléments <field-definition> et <relatedset-definition>, chacun contenant des attributs pour l'une des rubriques de ce jeu de résultats.

L'attribut <field-definition> définit :

- si la rubrique est de type `auto-enter` ("yes" ou "no")
- si la rubrique est de type `four-digit-year` ("yes" ou "no")
- si la rubrique est de type `Global` ("yes" ou "no")
- le nombre maximal de valeurs multivaluées (attribut `max-repeat`)
- le nombre maximal de caractères autorisé (attribut `max-characters`)
- si la rubrique est de type `not-empty` ("yes" ou "no")
- si la rubrique est réservée aux données numériques ("yes" ou "no")
- le format du résultat ("text", "number", "date", "time", "timestamp" ou "container")
- si la rubrique est de type `time-of-day` ("yes" ou "no")
- le type ("normal", "calculation" ou "summary")
- et le nom de la rubrique (entièrement qualifié si nécessaire)

L'élément <relatedset-definition> correspond à une table externe. Chaque rubrique liée dans une table externe est représentée par l'élément `fielddefinition` intégré à l'élément `relatedsetdefinition`. Si une table externe possède plusieurs rubriques liées, les définitions de ces rubriques sont regroupées dans un même élément `relatedsetdefinition`.

### élément <resultset>

L'élément <resultset> contient les éléments <record> renvoyés comme étant le résultat d'une requête et d'un attribut pour le nombre total d'enregistrements trouvés. Chaque élément <record> contient les données de rubrique pour l'un des enregistrements dans l'ensemble des résultats, y compris les attributs `mod-id` et `record-id` de l'enregistrement, et l'élément <data> contenant les données pour l'une des rubriques de l'enregistrement.

Chaque enregistrement d'une table externe est représenté par un élément <record> intégré à l'élément <relatedset>. L'attribut de comptage de l'élément <relatedset> indique le nombre d'enregistrements de la table externe, et l'attribut de table spécifie la table associée à la table externe.



## Données XML dans la grammaire fmresultset

### Exemple

```
<fmresultset xmlns="http://www.filemaker.com/xml/fmresultset" version="1.0">
  <error code="0"/>
  <product build="03/29/2019" name="FileMaker Web Publishing Engine"
    version="18.0.1.0"/>
  <datasource database="art" date-format="MM/dd/yyyy" layout="web3"
    table="art" time-format="HH:mm:ss" timestamp-format="MM/dd/yyyy HH:mm:ss"
    total-count="12"/>
  <metadata>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="Title" not-empty="no" numeric-only="no" result="text"
      time-of-day="no" type="normal"/>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="Artist" not-empty="no" numeric-only="no" result="text"
      time-of-day="no" type="normal"/>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="Style" not-empty="no" numeric-only="no" result="text"
      time-of-day="no" type="normal"/>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="length" not-empty="no" numeric-only="no"
      result="number" time-of-day="no" type="calculation"/>
    <relatedset-definition table="artlocations">
      <field-definition auto-enter="no" four-digit-year="no" global="no" max-
        repeat="1" name="artlocations::Location" not-empty="no" numeric-
        only="no" result="text" time-of-day="no" type="normal"/>
      <field-definition auto-enter="no" four-digit-year="no" global="no" max-
        repeat="1" name="artlocations::Date" not-empty="no" numeric-only="no"
        result="date" time-of-day="no" type="normal"/>
    </relatedset-definition>
  </metadata>
  <resultset count="1" fetch-size="1">
    <record mod-id="6" record-id="14">
      <field name="Title">
        <data>Spring in Giverny 3</data>
      </field>
      <field name="Artist">
        <data>Claude Monet</data>
      </field>
      <field name="Style">
        <data/>
      </field>
      <field name="length">
        <data>19</data>
      </field>
      <relatedset count="0" table="artlocations"/>
    </record>
  </resultset>
</fmresultset>
```

## Utilisation d'autres grammaires XML FileMaker

Les autres grammaires XML FileMaker contiennent des informations au sujet des types de rubrique, des listes de valeur et des modèles. `FMPXMLRESULT` est une grammaire fonctionnellement équivalente à `fmresultset`. Pour accéder aux listes de valeurs et aux informations d'affichage des rubriques dans des modèles, utilisez la grammaire `FMPXMLLAYOUT`. Les grammaires `FMPXMLRESULT` et `FMPXMLLAYOUT` sont plus compactes pour les échanges de données.

Pour utiliser la grammaire `FMPXMLRESULT`, indiquez le nom de la grammaire dans l'URL qui recherche le document XML à partir du moteur de publication Web :

```
FMPXMLRESULT.xml
```

### Exemple

```
http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?-db=employees&-lay=family
&-findall
```

Pour utiliser la grammaire `FMPXMLLAYOUT`, indiquez le nom de la grammaire avec la commande de requête `-view` dans l'URL qui recherche le document XML à partir du moteur de publication Web :

```
FMPXMLLAYOUT.xml
```

### Exemple

```
http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?-db=employees&-lay=family
&-view
```

**Remarque** Veillez à entrer le nom des grammaires `FMPXMLRESULT` et `FMPXMLLAYOUT` en majuscules.

Dans le document XML généré, le moteur de Publication Web référence la définition du type de document de la grammaire dans l'instruction `<!DOCTYPE>` située sur la deuxième ligne du document, immédiatement après l'instruction `<?xml . . . ?>`. L'instruction `<!DOCTYPE>` indique l'URL à employer pour télécharger la DTD relative à la grammaire.

## Description des éléments de la grammaire `FMPXMLRESULT`

Dans la grammaire `FMPXMLRESULT`, l'élément `<DATABASE>` contient les attributs `NAME`, `RECORDS`, `DATEFORMAT`, `LAYOUT` et `TIMEFORMAT`.

L'attribut `DATEFORMAT` de l'élément `<DATABASE>` permet de déterminer le format des dates dans le document XML. L'attribut `TIMEFORMAT` de l'élément `<DATABASE>` permet de déterminer le format des heures dans le document XML. Les formats date et heure des grammaires `FMPXMLRESULT` et `fmresultset` sont identiques. Consultez la section « Description des éléments de la grammaire `fmresultset` », page 31.

L'élément `<METADATA>` de la grammaire `FMPXMLRESULT` comporte un ou plusieurs éléments `<FIELD>` contenant chacun des informations pour l'une des rubriques/colonnes de l'ensemble des résultats, dont le nom de la rubrique définie dans la base de données, le type de rubrique, la tolérance du choix Oui ou Non pour les rubriques vides (attribut `EMPTYOK`), ainsi que le nombre maximal de rubriques multivaluées (attribut `MAXREPEAT`). Les valeurs autorisées pour les types de rubriques sont les suivantes : `TEXTE`, `NOMBRE`, `DATE`, `HEURE`, `HORODATAGE` et `CONTENEUR`.

L'élément `<RESULTSET>` contient tous les éléments `<ROW>` renvoyés comme étant le résultat d'une requête et d'un attribut pour le nombre total d'enregistrements trouvés. Chaque élément `<ROW>` contient les données de la rubrique/colonne pour l'une des rangées de l'ensemble des résultats. Ces données incluent `RECORDID` et `MODID` pour la rangée (voir « Paramètre de requête `-modid` (ID de la modification) », page 56) et l'élément `<COL>`. L'élément `<COL>` contient les données pour une rubrique/colonne de la rangée dans laquelle plusieurs éléments `<DATA>` représentent l'une des valeurs d'une rubrique multivaluée ou d'une rubrique de table externe.

## Données XML dans la grammaire FMPXMLRESULT

### Exemple

```
<FMPXMLRESULT xmlns="http://www.filemaker.com/fmpxmlresult">
  <ERRORCODE>0</ERRORCODE>
  <PRODUCT BUILD="03/29/2019" NAME="FileMaker Web Publishing Engine"
    VERSION="18.0.1.0"/>
  <DATABASE DATEFORMAT="MM/dd/yyyy" LAYOUT="web" NAME="art" RECORDS="12"
    TIMEFORMAT="HH:mm:ss"/>
  <METADATA>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Title" TYPE="TEXT"/>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Artist" TYPE="TEXT"/>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Image" TYPE="CONTAINER"/>
  </METADATA>
  <RESULTSET FOUND="1">
    <ROW MODID="7" RECORDID="4">
      <COL>
        <DATA>Village Market</DATA>
      </COL>
      <COL>
        <DATA>Camille Pissarro</DATA>
      </COL>
      <COL>
        <DATA>/fmi/xml/cnt/Untitled.pct?-db=art&-lay=web&-recid=4
          &-field=Image(1)
        </DATA>
      </COL>
    </ROW>
  </RESULTSET>
</FMPXMLRESULT>
```

L'ordre des éléments `<COL>` correspond à l'ordre des éléments `<FIELD>` dans l'élément `<METADATA>` : par exemple, si les rubriques « Title » et « Artist » sont répertoriées dans l'élément `<METADATA>`, « Village Market » puis « Camille Pissarro » apparaissent dans le même ordre dans les éléments `<RESULTSET>` et `<ROW>`.

## Description des éléments dans la grammaire FMPXMLLAYOUT

Dans la grammaire FMPXMLLAYOUT, l'élément `<LAYOUT>` contient le nom du modèle, le nom de la base de données et les éléments `<FIELD>` de chaque rubrique figurant dans le modèle correspondant de la base de données. Chaque élément `<FIELD>` décrit le type de style de la rubrique et contient l'attribut `VALUELIST` pour toute liste de valeurs associée à la rubrique.

L'élément `<VALUELISTS>` contient un ou plusieurs éléments `<VALUELIST>` pour chaque liste de valeurs répertoriée dans le modèle, chacun de ces éléments comportant le nom de la liste de valeurs et un élément `<VALUE>` pour chaque valeur de la liste.

En fonction des options sélectionnées dans la boîte de dialogue **Rubriques pour la liste de valeurs** de la base de données FileMaker Pro, l'élément `<VALUE>` dispose d'un attribut `DISPLAY` qui contient la valeur de la première rubrique uniquement, la valeur de la seconde rubrique uniquement ou les deux rubriques d'une liste de valeurs. Supposons que la première rubrique d'une liste de valeurs stocke le numéro d'ID d'un style d'art (par exemple « 100 ») et que la seconde rubrique affiche le nom associé au style (par exemple « Impressionism »). Voici un résumé du contenu de l'attribut `DISPLAY` lorsque les différentes combinaisons d'options sont sélectionnées dans la boîte de dialogue **Rubriques pour la liste de valeurs** :

- Si l'option **Afficher également les valeurs de la seconde rubrique** n'est pas sélectionnée, l'attribut `DISPLAY` contient pour valeur dans la première rubrique, d'une liste de valeurs uniquement.

### Exemple

L'attribut `DISPLAY` contient le numéro d'identifiant du style d'art uniquement :

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="100">100</VALUE>
    <VALUE DISPLAY="101">101</VALUE>
    <VALUE DISPLAY="102">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

- Si les options **Afficher également les valeurs de la seconde rubrique** et **Afficher uniquement les valeurs de la seconde rubrique** sont toutes les deux sélectionnées, l'attribut `DISPLAY` contient la valeur dans la seconde rubrique uniquement.

### Exemple

L'attribut `DISPLAY` contient le nom du style d'art uniquement :

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="Impressionism">100</VALUE>
    <VALUE DISPLAY="Cubism">101</VALUE>
    <VALUE DISPLAY="Abstract">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

- Si l'option **Afficher également les valeurs de la seconde rubrique** est sélectionnée et que l'option **Afficher uniquement les valeurs de la seconde rubrique** ne l'est pas, l'attribut DISPLAY contient les valeurs, dans les deux rubriques, d'une liste de valeurs.

### Exemple

L'attribut DISPLAY contient à la fois le numéro d'identifiant du style d'art et son nom :

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="100 Impressionism">100</VALUE>
    <VALUE DISPLAY="101 Cubism">101</VALUE>
    <VALUE DISPLAY="102 Abstract">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

Pour les rubriques de date, d'heure et d'horodatage, les données des listes de valeurs sont formatées à l'aide du format « fm » pour ce type de rubrique. Les formats fm sont `jj/mm/aaaa` pour la date, `hh:mm:ss` pour l'heure et `jj/mm/aaaa hh:mm:ss` pour l'horodatage. Par exemple, si une liste de valeur « birthdays » est utilisée pour un menu local dans la rubrique « birthdate » d'un modèle et que la rubrique « birthdate » est de type date, alors les valeurs résultantes pour cette liste de valeurs seront toutes au format de date « fm ».

**Remarque** Si deux rubriques de types différents dans un modèle partagent la même liste de valeurs, le type de la première rubrique détermine le format des données de la liste de valeurs.

## Données XML dans la grammaire FMPXMLLAYOUT

### Exemple

```
<FMPXMLLAYOUT xmlns="http://www.filemaker.com/fmpxmllayout">
  <ERRORCODE>0</ERRORCODE>
  <PRODUCT BUILD="03/29/2019" NAME="FileMaker Web Publishing Engine"
    VERSION="18.0.1.0"/>
  <LAYOUT DATABASE="art" NAME="web2">
    <FIELD NAME="Title">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="Artist">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="Image">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="artlocations::Location">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="artlocations::Date">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="Style">
      <STYLE TYPE="POPUPMENU" VALUELIST="style"/>
    </FIELD>
  </LAYOUT>
  <VALUELISTS>
    <VALUELIST NAME="style">
      <VALUE DISPLAY="Impressionist">Impressionist</VALUE>
      <VALUE DISPLAY="Modern">Modern</VALUE>
      <VALUE DISPLAY="Abstract">Abstract</VALUE>
    </VALUELIST>
  </VALUELISTS>
</FMPXMLLAYOUT>
```

## À propos des données UTF-8

Les données XML générées par le moteur de publication Web sont codées au format UTF-8 (Unicode Transformation Format 8 bits). Ce format compresse les données du format Unicode standard 16 bits au format 8 bits pour les caractères ASCII. Les analyseurs XML sont nécessaires à la prise en charge des codages Unicode et UTF-8.

Le codage au format UTF-8 comprend les représentations directes de la plupart des caractères utilisés en anglais, avec les valeurs 0 à 127 pour le jeu de caractères ASCII standard, et fournit des codages multioctets pour les caractères Unicode dont les valeurs sont supérieures.

**Remarque** Veillez à utiliser un navigateur Web ou un éditeur de texte prenant en charge les fichiers UTF-8.

Le format du codage UTF-8 présente les caractéristiques suivantes :

- Tous les caractères ASCII sont des caractères UTF-8 à un octet. Une chaîne ASCII valide est une chaîne UTF-8 valide.
- Tout caractère non ASCII (tout caractère dont le bit de poids fort est défini) fait partie d'un caractère multioctet.
- Le premier octet d'un caractère au format UTF-8 indique le nombre d'octets supplémentaires dans le caractère en question.
- Le premier octet d'un caractère multioctet est facilement identifiable par rapport aux octets suivants. Il est ainsi facile de repérer le début d'un caractère quel que soit son emplacement dans un flux de données.
- La conversion entre le format UTF-8 et le format Unicode ne pose aucun problème.
- Le codage au format UTF-8 est relativement compact. Dans le cas d'un texte contenant un pourcentage élevé de caractères ASCII, ce type de codage est plus compact que le codage au format Unicode. Dans le pire des cas, une chaîne UTF-8 n'est que 50 % plus grande que la chaîne Unicode équivalente.

## Utilisation de chaînes de requête FileMaker pour rechercher des données XML

Pour rechercher des données XML dans une base de données FileMaker Pro, utilisez les commandes et les paramètres de requête dans une chaîne de requête.

### Exemple

Utilisez la commande de requête `-findall` dans la chaîne de requête suivante d'une URL pour rechercher une liste de tous les produits d'une base de données FileMaker Pro dont le nom est « products » :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=products  
&-lay=sales&-findall
```

Une chaîne de requête doit contenir uniquement une seule commande de requête, du type `-new`. La plupart des commandes de requête exigent également plusieurs paramètres de requête concordants dans la chaîne de requête. Par exemple, toutes les commandes de requête, à l'exception de `-dbnames`, ont besoin du paramètre `-db`, qui spécifie la base de données sur laquelle effectuer la recherche.

Vous pouvez également utiliser des commandes et des paramètres de requête dans une URL.

Cette section contient un récapitulatif des commandes et des paramètres de requête de FileMaker. Pour en savoir plus sur leur utilisation dans une chaîne de requête, consultez la section « Noms corrects utilisés dans les chaînes de requête XML », page 43.

Utilisez cette commande de requête	Pour exécuter cette commande
-dbnames	Extraire le nom de toutes les bases de données hébergées et partagées sur le Web.
-delete	Supprimer l'enregistrement.
-dup	Dupliquer l'enregistrement.
-edit	Modifier un enregistrement.
-find	Rechercher un ou plusieurs enregistrements.
-findall	Rechercher tous les enregistrements.
-findany	Rechercher un enregistrement aléatoire.
-findquery	Accomplir une requête de recherche complexe ou composée.
-layoutnames	Extraire le nom de tous les modèles disponibles correspondant à une base de données spécifique, hébergée et partagée sur le Web.
-new	Ajouter un nouvel enregistrement.
-scriptnames	Extraire le nom de tous les scripts disponibles correspondant à une base de données spécifique, hébergée et partagée sur le Web.
-view	Extraire les informations du modèle d'une base de données si la grammaire <code>FMPXMLLAYOUT</code> est spécifiée. Extraire la section de <code>&lt;metadata&gt;</code> du document XML et un jeu d'enregistrements vides si la grammaire <code>fmresultset</code> ou <code>FMPXMLRESULT</code> est spécifiée.

Utilisez ce paramètre de requête	Avec ces commandes de requête
-db (nom de la base de données)	Obligatoire pour toutes les commandes de requête, à l'exception de <code>-dbnames</code>
-delete.related	Facultatif pour <code>-edit</code>
-field	Obligatoire pour indiquer une rubrique dans une URL pour les requêtes de type Conteneur. Consultez la section « À propos de la syntaxe d'URL pour les objets FileMaker Pro de type Conteneur dans les solutions XML », page 27.
fieldname	Un nom de rubrique au moins est obligatoire avec <code>-edit</code> . Facultatif pour <code>-find</code> . Consultez la section « Paramètre de requête <code>fieldname</code> (nom de rubrique non-Conteneur) », page 53.
fieldname.op (opérateur)	Facultatif pour <code>-find</code>
-lay (nom du modèle)	Obligatoire pour toutes les commandes de requête, à l'exception de <code>-dbnames</code> , <code>-layoutnames</code> et <code>-scriptnames</code>
-lay.response (passer d'un modèle à un modèle pour obtenir une réponse XML)	Obligatoire pour toutes les commandes de requête, à l'exception de <code>-dbnames</code> , <code>-layoutnames</code> et <code>-scriptnames</code>
-lop (opérateur logique)	Facultatif pour <code>-find</code>
-max (nombre maximal d'enregistrements)	Facultatif pour <code>-find</code> , <code>-findall</code> et <code>-findquery</code>
-modid (ID de modification)	Facultatif pour <code>-edit</code>
-query	Obligatoire pour les requêtes de recherche composée <code>-findquery</code>
-recid (ID de l'enregistrement)	Obligatoire pour <code>-edit</code> , <code>-delete</code> , <code>-dup</code> . Facultatif pour <code>-find</code>
-relatedsets.filter	Facultatif pour <code>-find</code> , <code>-findall</code> , <code>-findany</code> , <code>-edit</code> , <code>-new</code> , <code>-dup</code> et <code>-findquery</code>



Utilisez ce paramètre de requête	Avec ces commandes de requête
<code>-relatedsets.max</code>	Facultatif avec <code>-find</code> , <code>-edit</code> , <code>-new</code> , <code>-dup</code> et <code>-findquery</code> .
<code>-script</code> (exécution du script)	Facultatif pour <code>-find</code> , <code>-findall</code> , <code>-findany</code> , <code>-new</code> , <code>-edit</code> , <code>-delete</code> , <code>-dup</code> , <code>-view</code> et <code>-findquery</code>
<code>-script.param</code> (transmettre une valeur de paramètre au script spécifié par <code>-script</code> )	Facultatif pour <code>-script</code> et <code>-findquery</code>
<code>-script.prefind</code> (exécution du script avant <code>-find</code> , <code>-findany</code> et <code>-findall</code> )	Facultatif pour <code>-find</code> , <code>-findany</code> , <code>-findall</code> et <code>-findquery</code>
<code>-script.prefind.param</code> (transmettre une valeur de paramètre au script spécifié par <code>-script.prefind</code> )	Facultatif pour <code>-script.prefind</code> et <code>-findquery</code>
<code>-script.presort</code> (exécuter le script avant le tri)	Facultatif pour <code>-find</code> , <code>-findall</code> et <code>-findquery</code>
<code>-script.presort.param</code> (transmettre une valeur de paramètre au script spécifié par <code>-script.presort</code> )	Facultatif pour <code>-script.presort</code> et <code>-findquery</code>
<code>-skip</code> (ignorer les enregistrements)	Facultatif pour <code>-find</code> , <code>-findall</code> et <code>-findquery</code>
<code>-sortfield.[1-9]</code> (trier la rubrique)	Facultatif pour <code>-find</code> , <code>-findall</code> et <code>-findquery</code>
<code>-sortorder.[1-9]</code> (trier l'ordre)	Facultatif pour <code>-find</code> , <code>-findall</code>

## Passage d'un modèle à un autre pour obtenir une réponse XML

Le paramètre de requête `-lay` indique le modèle à utiliser pour extraire des données XML. Souvent, le même modèle se révèle adapté au traitement des données répertoriées dans la requête. Dans certains cas, vous devrez rechercher les données qui utilisent un modèle contenant des rubriques qui, pour des raisons de sécurité, n'existent dans aucun autre modèle servant à l'affichage des résultats. (Pour rechercher des données dans une rubrique, il faut que celle-ci soit placée dans le modèle indiqué dans la requête XML.)

Pour définir un modèle destiné à l'affichage des réponses XML qui soit différent du modèle utilisé pour le traitement de la requête XML, utilisez le paramètre de requête facultatif `-lay.response`.

### Exemple

La requête suivante recherche les valeurs supérieures à 100 000 dans la rubrique « Salary » du modèle « Budget ». Les données obtenues s'affichent au moyen d'un modèle « ExecList », qui n'inclut pas la rubrique « Salary ».

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=Budget&Salary=100000&Salary.op=gt&-find&-lay.response=ExecList
```

## Comprendre le traitement d'une requête XML

Plusieurs paramètres de requête interviennent dans le traitement d'une requête XML et dans la génération d'un document XML.

Voici l'ordre dans lequel FileMaker Server et le moteur de publication Web traitent une requête XML :

1. Traitez le paramètre de requête `-lay`.
2. Définissez les valeurs des rubriques de type Global dans la requête (la partie `.global=` d'une URL).
3. Traitez le paramètre de requête `-script.prefind` si indiqué.
4. Traitez les commandes de requête de type `-find` ou `-new`.
5. Traitez le paramètre de requête `-script.presort`, si indiqué.
6. Triez les données obtenues, si un tri a été indiqué.
7. Traitez le paramètre de requête `-script`, si indiqué.
8. Traitez le paramètre de requête `-lay.response` pour passer d'un modèle à un autre, si indiqué.
9. Générez le document XML.

Si l'une des étapes décrites ci-dessus génère un code d'erreur, le traitement de la requête s'arrête ; les étapes suivantes ne sont alors pas exécutées. Toutefois, toutes les étapes antérieures dans la requête restent exécutées.

Par exemple, imaginons une requête qui supprime l'enregistrement ouvert, trie les enregistrements, puis exécute un script. Si le paramètre `-sortfield` pointe vers une rubrique qui n'existe pas, la requête supprime l'enregistrement ouvert et renvoie le code d'erreur 102 (Rubrique manquante), sans toutefois exécuter le script.

## Dépannage de l'accès aux documents XML

Si vous rencontrez des difficultés pour accéder aux documents XML avec le moteur de publication Web, vérifiez les points suivants :

- Les privilèges étendus contenus dans la base de données sont définis pour une Publication Web personnalisée pour XML et affectés à un compte utilisateur. Consultez la section « Activation de la Publication Web personnalisée dans une base de données », page 13.
- La base de données est hébergée sur le composant Serveur de bases de données de FileMaker Server, et elle est ouverte par FileMaker Server. Consultez l'[Aide FileMaker Server](#).
- Le nom de compte et le mot de passe que vous utilisez éventuellement pour la base de données sont corrects.
- Le composant serveur Web du déploiement FileMaker Server est en cours d'exécution.
- Le composant moteur de publication Web du déploiement FileMaker Server est en cours d'exécution.
- La publication XML est activée à l'aide de l'interface de ligne de commande (CLI). Consultez l'[Aide FileMaker Server](#).

# Chapitre 5

## Noms corrects utilisés dans les chaînes de requête XML

Ce chapitre décrit les noms valides des commandes et des paramètres de requête que vous pouvez utiliser dans une chaîne de requête XML lorsque vous accédez à des données FileMaker à l'aide du moteur de publication Web.

### À propos des commandes et des paramètres de requête

Voici une liste complète des noms de commandes de requête et des noms de paramètres de requête :

Noms des commandes de requête	Noms des paramètres de requête
-dbnames (Voir page 49.)	-db (Voir page 52.)
-delete (Voir page 49.)	-field (Voir page 53.)
-dup (Voir page 49.)	fieldname (Voir page 53.)
-edit (Voir page 49.)	fieldname.op (Voir page 54.)
-find, -findall, -findany (Voir page 50.)	-lay (Voir page 55.)
-findquery (Voir page 50.)	-lay.response (Voir page 55.)
-layoutnames (Voir page 51.)	-lop (Voir page 56.)
-new (Voir page 51.)	-max (Voir page 56.)
-scriptnames (Voir page 52.)	-modid (Voir page 56.)
-view (Voir page 52.)	-query (Voir page 57.)
	-recid (Voir page 58.)
	-relatedsets.filter (Voir page 58.)
	-relatedsets.max (Voir page 59.)
	-script (Voir page 60.)
	-script.param (Voir page 60.)
	-script.prefind (Voir page 60.)
	-script.prefind.param (Voir page 61.)
	-script.presort (Voir page 61.)
	-script.presort.param (Voir page 61.)
	-skip (Voir page 62.)
	-sortfield.[1-9] (Voir page 62.)
	-sortorder.[1-9] (Voir page 63.)

**Important** Le paramètre `-lay` servant à spécifier un modèle est nécessaire avec toutes les commandes de requête, à l'exception de `-dbnames`, `-layoutnames` et `-scriptnames`.

## Consignes d'utilisation des commandes et des paramètres de requête

Lorsque vous utilisez des commandes et des paramètres de requête dans une chaîne de requête, gardez les consignes suivantes à l'esprit :

- Une chaîne de requête doit contenir une seule commande de requête, ni plus, ni moins. Une chaîne de requête peut par exemple contenir `-new` pour ajouter un nouvel enregistrement, mais `-new` et `-edit` ne peuvent être contenus dans la même chaîne de requête.
- La majorité des commandes de requête nécessitent divers paramètres de requêtes correspondants dans la chaîne de requête. Par exemple, toutes les commandes de requête, à l'exception de `-dbnames`, ont besoin du paramètre `-db`, qui spécifie la base de données sur laquelle effectuer la recherche. Consultez le tableau des paramètres requis dans la section « Utilisation de chaînes de requête FileMaker pour rechercher des données XML », page 39.
- Pour les paramètres de requête et les noms de rubrique, indiquez la valeur en particulier que vous souhaitez utiliser, par exemple `-db=employees`. Pour les commandes de requête, n'indiquez pas le signe `=` ou une valeur après le nom de la commande, par exemple `-findall`.
- Le moteur de publication Web convertit tous les mots réservés en minuscules, notamment les commandes de requête, les paramètres de requête et les valeurs de commande pour lesquelles des valeurs spécifiques sont attendues (par exemple : `-lop=and`, `-lop=or`, `-sortorder=ascend`, `-sortorder=descend`, `-max=all`).
- Les noms de bases de données, de modèles et de rubriques utilisés dans les chaînes de requête ne tiennent pas compte de la casse, vous pouvez par exemple indiquer `-lay=mylayout` pour spécifier le nom de modèle `MyLayout`.
- Il est déconseillé d'utiliser des points ou des parenthèses dans les noms de rubrique. Dans certains cas, les noms de rubrique comportant des points peuvent fonctionner mais en aucun cas, ils ne peuvent contenir les exceptions suivantes :
  - Le point ne peut pas être suivi d'un chiffre. Par exemple, `myfield.9` est un nom de rubrique invalide.
  - Le point ne peut pas être suivi de la chaîne de texte `op` (les deux lettres « op »). Par exemple, `myfield.op` est un nom de rubrique invalide.
  - Le point ne peut pas être suivi de la chaîne de texte `global` (le mot « global »). Par exemple, `myfield.global` est un nom de fichier invalide.Les noms de rubrique contenant l'une de ces exceptions ne sont pas accessibles via XML lorsque vous utilisez une requête HTTP. Ces constructions sont destinées aux ID d'enregistrements, comme décrit dans la section « A propos de la syntaxe d'un nom de rubrique entièrement qualifié », page 46.
- Pour la commande `-find`, la valeur d'une rubrique ne tient pas compte de la casse. Par exemple, vous pouvez aussi bien écrire `Field1=Blue` que `Field1=blue`. Pour les commandes `-new` and `-edit`, la casse que vous utilisez dans la valeur d'une rubrique est conservée et stockée dans la base de données exactement comme vous la spécifiez dans la chaîne de requête. Par exemple, `LastName=Doe`.

## Analyse de commande de requête

Le moteur de publication Web analyse les commandes de requête dans l'ordre suivant, et arrête d'analyser les requêtes XML à la première erreur. Si un code d'erreur est renvoyé, ce dernier correspond à la première erreur détectée.

### 1. La requête contient-elle une commande, et cette commande est-elle valide ?

Il s'agit d'une erreur si la requête ne contient pas de commande ou utilise une commande inconnue.

#### Exemple

```
-database
```

### 2. La requête comporte-t-elle deux commandes ?

#### Exemple

```
-find&-edit
```

### 3. La requête comporte une valeur incorrecte pour une commande ou un paramètre ?

#### Exemple

```
-lop=amd
```

### 4. Le paramètre de nom de base de données requis (`-db` parameter) manque-t-il dans la requête ?

### 5. Le paramètre de nom de modèle requis (`-lay` parameter) manque-t-il dans la requête ?

### 6. La requête comporte-t-elle un tri non valide ?

### 7. La requête contient-elle des paramètres de rubrique non valides ?

**Remarque** Si une requête contient une information valide, mais superflue, elle est traitée sans erreur. Par exemple, si vous spécifiez le paramètre `-lop` dans une commande `-delete`, ce paramètre `-lop` est ignoré, car il ne rend pas la requête non valide ou ambiguë.

Pour plus d'informations sur les codes d'erreur spécifiques renvoyés, consultez l'annexe A, « Codes d'erreur de la Publication Web personnalisée ».

## A propos de la syntaxe d'un nom de rubrique entièrement qualifié

Un nom de rubrique entièrement qualifié identifie une instance précise d'une rubrique. Dans la mesure où les rubriques possédant des noms communs peuvent être basées sur différentes tables, vous devez utiliser des noms entièrement qualifiés, dans certains cas, pour éviter les erreurs.

La syntaxe permettant de spécifier un nom de rubrique entièrement qualifié est la suivante :

```
table-name::field-name(repetition-number).record-id
```

où :

- `table-name` est le nom de la table qui contient la rubrique. Le nom de la table est uniquement requis si la rubrique ne se trouve pas dans la table sous-jacente du modèle spécifié dans la chaîne de requête.
- `field-name(repetition-number)` est une valeur spécifique dans une rubrique multivaluée, uniquement requise pour les rubriques multivaluées. Le nombre de répétitions commence à compter à partir du chiffre 1. Par exemple, `field-name(2)` correspond à la seconde valeur dans la rubrique multivaluées. Si vous n'indiquez pas de numéro de valeur pour une rubrique multivaluée, c'est la première valeur qui est utilisée. Le numéro de valeur est requis pour les commandes de requête `-new` and `-edit` impliquant des rubriques multivaluées, mais n'est pas requis pour la commande `-find`.
- `record-id` est l'ID de l'enregistrement, qui est uniquement requis si vous utilisez une chaîne de requête pour ajouter ou modifier des enregistrements dans des rubriques de table externe. Consultez les commandes de requête `-new` et `-edit` impliquant sections « Ajout d'enregistrements à une table externe » et « Modification d'enregistrements dans une table externe ». Le paramètre `record-id` est requis pour les commandes de requête `-new` et `-edit` impliquant des rubriques de table externe, mais n'est pas requis pour la commande `-find`.

**Remarque** Pour être accessibles, les rubriques doivent être placées dans le modèle que vous spécifiez dans la chaîne de requête.

## Utilisation de commandes de requêtes avec des rubriques de table externe

Les sections suivantes décrivent le fonctionnement des commandes de requêtes avec des rubriques de table externe.

### Ajout d'enregistrements à une table externe

Pour ajouter un nouvel enregistrement à une table externe en même temps qu'un enregistrement parent, utilisez la commande de requête `-new` et effectuez les opérations suivantes dans une chaîne de requête :

- Utilisez le nom de rubrique entièrement qualifié pour la rubrique de table externe correspondante.
- Spécifiez 0 en guise d'ID d'enregistrement après le nom de la rubrique de table externe liée.
- Spécifiez au moins une des rubriques pour l'enregistrement parent, avant de spécifier la rubrique de table externe liée.
- Spécifiez les données pour la rubrique source (rubrique clé) dans l'enregistrement parent.

## Exemple

L'URL suivante ajoute simultanément un nouvel enregistrement « Employee » parent pour John Doe et un nouvel enregistrement lié pour Jane dans la table externe. Le nom de la table liée est Dependents et le nom de la rubrique liée dans la table externe est Names. La rubrique source ID stocke un numéro d'identification de type Employee.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&FirstName=John&LastName=Doe&ID=9756&Dependents::Names.0=Jane&-new
```

**Remarque** Vous ne pouvez ajouter qu'un enregistrement correspondant à une table externe par requête.

## Modification d'enregistrements dans une table externe

Pour modifier un ou plusieurs enregistrements dans une table externe, utilisez la commande `-edit` et un ID d'enregistrement pour spécifier l'enregistrement parent qui contient les enregistrements de table externe que vous souhaitez modifier. Spécifiez l'enregistrement de table externe spécifique à modifier en utilisant son ID d'enregistrement dans un nom de rubrique entièrement qualifié. Vous pouvez déterminer un ID d'enregistrement à partir de l'attribut d'ID d'enregistrement de l'élément `<record>` dans l'élément `<relatedset>` dans les données XML. Consultez la section « Utilisation de la grammaire fmresultset », page 30.

## Exemples

L'URL suivante modifie un enregistrement d'une table externe dans laquelle l'enregistrement parent affiche l'ID d'enregistrement 1001. « Dependents » est le nom de la table liée, « Names » est le nom de la rubrique liée dans la table externe et le chiffre 2 dans « Names.2 » est l'ID d'enregistrement d'un enregistrement de table externe.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&Dependents::Names.2=Kevin&-edit
```

Comment utiliser une seule requête pour modifier plusieurs enregistrements de table externe par le biais de l'enregistrement parent :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&Dependents::Names.2=Kevin&Dependents::Names.5=Susan&-edit
```

Vous pouvez également utiliser la commande `-edit` et spécifier 0 en guise d'ID d'enregistrement de table externe pour ajouter un nouvel enregistrement lié dans la table externe pour un enregistrement parent existant.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&Dependents::Names.0=Timothy&-edit
```

## Suppression d'enregistrements dans une table externe

Pour supprimer des enregistrements de table externe, utilisez le paramètre `-delete.related` avec la commande `-edit` au lieu d'employer la commande `-delete`.

## Exemples

L'URL suivante supprime l'enregistrement « 1001 » de la table « employees » :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&-delete
```

En revanche, l'URL suivant supprime l'enregistrement de table externe dont l'ID est « 3 » de la table liée nommée « Dependents », et dont l'ID d'enregistrement parent est « 1001 ».

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&-delete.related=Dependents.3&-edit
```

Consultez la section « Paramètre de requête `-delete.related` (suppression d'enregistrements de la table externe) », page 53.

## Requêtes de rubriques de table externe

Dans une solution comportant de nombreux enregistrements liés, l'interrogation et le tri des enregistrements externes peuvent demander beaucoup de temps. Pour limiter le nombre d'enregistrements et de lignes à afficher dans un ensemble lié, utilisez les paramètres `-relatedsets.filter` et `-relatedsets.max` avec les requêtes de recherche. Consultez les sections « Paramètre de requête `-relatedsets.filter` (filtrage d'enregistrements de table externe) », page 58 et « Paramètre de requête `-relatedsets.max` (limitation d'enregistrements de table externe) », page 59.

## À propos de la syntaxe de spécification d'une rubrique de type Global

La syntaxe de spécification d'une rubrique de type Global est la suivante :

```
table-name::field-name(repetition-number).global
```

où `global` identifie une rubrique utilisant le stockage global. Pour en savoir plus sur `table-name` et `field-name(repetition-number)`, consultez la section « A propos de la syntaxe d'un nom de rubrique entièrement qualifié », page 46. Pour en savoir plus sur les rubriques de type Global, consultez l'[Aide FileMaker Pro](#).

Vous devez utiliser la syntaxe `.global` pour identifier une rubrique de type Global dans une chaîne de requête. Le moteur de publication Web définit les valeurs de paramètres pour les rubriques de type Global avant d'exécuter la commande de requête ou de définir toute autre valeur de paramètre dans la chaîne de requête. Pour les requêtes XML directes, les valeurs globales expirent immédiatement une fois la requête lancée.

Si vous n'utilisez pas la syntaxe `.global` pour identifier une rubrique de type Global dans une chaîne de requête, le moteur de publication Web traite la rubrique de type Global avec le reste de la chaîne de requête, sans définir la valeur de la rubrique de type Global au préalable.

## Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&Country.global=USA&-recid=1&-edit
```

## Références des commandes de requête

Cette section contient des informations sur des commandes de requête disponibles pour les requêtes XML.



### Commande de requête `-dbnames` (noms des bases de données)

Extrait le nom de toutes les bases de données hébergées par FileMaker Server et activées pour la Publication Web personnalisée avec XML.

**Paramètres de requête obligatoires :** (aucun)

#### Exemple

Pour extraire les noms de bases de données :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-dbnames
```

### Commande de requête `-delete` (supprimer l'enregistrement)

Supprime l'enregistrement indiqué par le paramètre `-recid`.

**Paramètres de requête obligatoires :** `-db`, `-lay`, `-recid`

**Paramètres de requête facultatifs :** `-script`

#### Exemple

Pour supprimer un enregistrement :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=4&-delete
```

### Commande de requête `-dup` (dupliquer l'enregistrement)

Duplique l'enregistrement indiqué par le paramètre `-recid`.

**Paramètres de requête obligatoires :** `-db`, `-lay`, `-recid`

**Paramètres de requête facultatifs :** `-script`

#### Exemple

Pour dupliquer l'enregistrement spécifié :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=14&-dup
```

### Commande de requête `-edit` (modifier l'enregistrement)

Met à jour l'enregistrement indiqué par le paramètre `-recid` et renseigne les rubriques avec le contenu de n'importe quelle paire nom rubrique/valeur. Le paramètre `-recid` indique le nom de l'enregistrement qui doit être modifié.

**Paramètres de requête obligatoires :** `-db`, `-lay`, `-recid` et un ou plusieurs noms de rubriques

**Paramètres de requête facultatifs :** `-modid`, `-script`, nom de rubrique

**Remarque** La commande `-edit` peut servir à modifier des enregistrements dans une table externe. Consultez la section « Modification d'enregistrements dans une table externe », page 47.

#### Exemple

Pour modifier un enregistrement :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=13&Country=USA&-edit
```

## Commandes de requête -find, -findall ou -findany (rechercher les enregistrements)

Soumet une requête utilisant des critères de recherche définis.

**Paramètres de requête obligatoires :** -db, -lay

**Paramètres de requête facultatifs :** -recid, -lop, -op, -max, -skip, -sortorder, -sortfield, -script, -script.prefind, -script.presort, field name

### Exemples

Pour chercher un enregistrement par son nom de rubrique :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=family&Country=USA&-find
```

Pour chercher un enregistrement par son ID d'enregistrement :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=427&-find
```

Pour chercher tous les enregistrements de la base de données, utilisez -findall :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-findall
```

Pour chercher un enregistrement de façon aléatoire, utilisez -findany :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-findany
```

### Remarques

- Le fait de spécifier un nom de rubrique à plusieurs reprises dans une même requête n'est pas pris en charge ; FileMaker Server analyse toutes les valeurs mais utilise uniquement la dernière valeur analysée.
- Pour éviter de surcharger la mémoire de l'ordinateur si vous utilisez la commande -findall, indiquez par défaut un nombre maximum d'enregistrements à renvoyer par page. Pour cela, utilisez le paramètre de requête -max.

## Commande de requête -findquery (recherche composée)

Soumet une requête de recherche avec des critères de recherche multiples ainsi que des exceptions.

**Paramètres de requête obligatoires :** -db, -lay, -query

**Paramètres de requête facultatifs :** -max, -skip, -sortorder, -sortfield, -script, -script.prefind, -script.presort

### Exemple

Rechercher les enregistrements des chats ou des chiens qui ne s'appellent pas « Fluffy » :

```
http://host/fmi/xml/fmresultset.xml?-db=vetclinic&-lay=animals
&-query=(q1);(q2);!(q3)&-q1=typeofanimal&-q1.value=Cat&-q2=typeofanimal
&-q2.value=Dog&-q3=name&-q3.value=Fluffy&-findquery
```

### Utilisation de la commande `-findquery` pour les recherches composées

Une instruction `-findquery` se compose de quatre parties, dans l'ordre suivant :

- Le paramètre de requête `-query`.
- Les déclarations de demande de requête, composées de l'identificateur de requête et des opérations de demande.
- Le champ de recherche et les définitions de valeur pour chaque identificateur de requête.
  - Définissez les identifiants de requête. Un identificateur de requête est représenté par la lettre « q » suivie d'un nombre. Par exemple : `-q1`
  - Définissez les valeurs d'identifiant avec le paramètre. Par exemple :  
`-q1.value=fieldvalue`
  - Définissez les opérateurs d'identifiant de requête en l'incluant en tant que partie de l'expression `fieldvalue`. Par exemple, utiliser un astérisque comme opérateur « begins with » : `-q1.value=fieldvalue*`
- La commande `-findquery`, à la fin de l'instruction complète.

Pour en savoir plus sur l'utilisation du paramètre `-query`, consultez la section « Paramètre de requête `-query` (requête de recherche composée) », page 57.

### Commande de requête `-layoutnames` (noms des modèles)

Extrait le nom de tous les modèles disponibles pour une base de données spécifique, hébergée par FileMaker Server et activée pour la Publication Web personnalisée avec XML.

**Paramètres de requête obligatoires :** `-db`

#### Exemple

Pour extraire les noms des modèles disponibles :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-layoutnames
```

### Commande de requête `-new` (nouvel enregistrement)

Crée un nouvel enregistrement et renseigne cet enregistrement avec le contenu de n'importe quelle paire nom/valeur d'une rubrique.

**Paramètres de requête obligatoires :** `-db`, `-lay`

**Paramètre de requête facultatif :** un ou plusieurs noms de rubriques, `-script`

**Remarque** Pour plus d'informations sur la manière d'inclure de nouvelles données pour une table externe, consultez la section « Ajout d'enregistrements à une table externe », page 46.

#### Exemple

Pour ajouter un nouvel enregistrement :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&Country=Australia&-new
```

## Commande de requête `-scriptnames` (noms des scripts)

Extrait le nom de tous les scripts disponibles pour une base de données spécifique, hébergée par FileMaker Server et activée pour la Publication Web personnalisée avec XML.

**Paramètres de requête obligatoires :** `-db`

### Exemple

Pour extraire les noms de tous les scripts :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-scriptnames
```

## Commande de requête `-view` (affichage des informations de modèle)

Si la grammaire `FMPXMLLAYOUT` est spécifiée, cette commande extrait des informations de modèle d'une base de données et les affiche dans la grammaire `FMPXMLLAYOUT`. Si une grammaire de données (`fmresultset` or `FMPXMLRESULT`) est spécifiée, la commande extrait la section de métadonnées d'un document XML et un jeu d'enregistrements vide.

**Paramètres de requête obligatoires :** `-db`, `-lay`

**Paramètres de requête facultatifs :** `-script`

### Exemples

Pour extraire des informations de modèle :

```
http://192.168.123.101/fmi/xml/FMPXMLLAYOUT.xml?-db=employees  
&-lay=departments&-view
```

Pour extraire des informations de métadonnées :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-view
```

## Références de paramètre de requête

Cette section contient des informations sur des paramètres de requête disponibles pour les requêtes XML.

### Paramètre de requête `-db` (nom de base de données)

Indique la base de données à laquelle la commande de requête s'applique.

**Valeur :** Nom de la base de données sans l'extension de nom de fichier, si elle existe.

**Remarque** Lorsque vous indiquez le nom de la base de données pour le paramètre `-db` dans les chaînes de requête, n'incluez pas l'extension du nom de fichier. Le nom de fichier réel de la base de données peut éventuellement comprendre une extension, mais les extensions ne sont pas autorisées comme valeur pour le paramètre `-db`.

**Obligatoire pour :** Toutes les commandes de requête, à l'exception de `-dbnames`

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-findall
```

### Paramètre de requête `-delete.related` (suppression d'enregistrements de la table externe)

Supprime un enregistrement dans une rubrique de table externe.

**Facultatif pour :** la commande de requête `-edit`

**Nécessite :** Un nom de table liée et un ID d'enregistrement

#### Exemple

L'exemple suivant supprime l'enregistrement de table externe dont l'ID est « 20 » de la table liée nommée « jobtable », et dont l'ID d'enregistrement parent est « 7 ».

```
http://host/fmi/xml/fmresultset.xml?-db=career&-lay=applications&-recid=7
&-delete.related=jobtable.20&-edit
```

### Paramètre de requête `-field` (nom de rubrique Conteneur)

Indique le nom d'une rubrique Conteneur.

**Obligatoire pour :** les requêtes de données dans une rubrique Conteneur.

Consultez la section « À propos de la syntaxe d'URL pour les données XML et les objets de type Conteneur », page 27.

### Paramètre de requête `fieldname` (nom de rubrique non-Conteneur)

Les noms de rubrique sont utilisés pour contrôler les critères de la commande de requête `-find` ou pour modifier le contenu d'un enregistrement. Quand vous devez spécifier une valeur pour une rubrique non-conteneur pour une commande ou un paramètre de requête, utilisez le nom de rubrique sans le trait d'union (`-`) en guise de partie nom de la paire nom/valeur.

**Nom :** Nom de la rubrique dans la base de données FileMaker Pro. Si la rubrique ne se trouve pas dans la table sous-jacente du modèle spécifié dans la chaîne de requête, le nom de rubrique doit être entièrement qualifié.

Il est déconseillé d'utiliser des points ou des parenthèses dans les noms de rubrique. Dans certains cas, les noms de rubrique comportant des points peuvent fonctionner mais en aucun cas, ils ne peuvent contenir les exceptions suivantes :

- Le point ne peut pas être suivi d'un chiffre. Par exemple, `myfield.9` est un nom de rubrique invalide.
- Le point ne peut pas être suivi de la chaîne de texte `op` (les deux lettres « op »). Par exemple, `myfield.op` est un nom de rubrique invalide.
- Le point ne peut pas être suivi de la chaîne de texte `global` (le mot « global »). Par exemple, `myfield.global` est un nom de fichier invalide.

Les noms de rubrique contenant l'une de ces exceptions ne sont pas accessibles via XML lorsque vous utilisez une requête HTTP. L'utilisation de points dans les noms de rubrique doit être réservée aux identifiants d'enregistrement, comme décrit dans la section « A propos de la syntaxe d'un nom de rubrique entièrement qualifié », page 46.

**Valeur :** pour les commandes de requête `-new` and `-edit`, indiquez la valeur que vous souhaitez stocker dans la rubrique de l'enregistrement actuel. Pour les commandes de requête `-find`, indiquez la valeur que vous souhaitez rechercher dans la rubrique. Lorsque vous indiquez la valeur d'une rubrique de type date, heure ou horodatage, spécifiez-la en utilisant le format « fm ». Les formats fm sont `jj/mm/aaaa` pour la date, `hh:mm:ss` pour l'heure et `jj/mm/aaaa hh:mm:ss` pour l'horodatage.

**Obligatoire pour :** la commande de requête `-edit`

**Facultatif pour :** les commandes de requête `-new` et `-find`

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-op=eq&FirstName=Sam&-max=1&-find
```

**Remarque** Le fait de spécifier un nom de rubrique à plusieurs reprises dans une même requête n'est pas pris en charge ; FileMaker Server analyse toutes les valeurs mais utilise uniquement la dernière valeur analysée.

### Paramètre de requête `fieldname.op` (opérateur de comparaison)

Indique l'opérateur de comparaison à appliquer au nom de rubrique précédant l'opérateur. Les opérateurs de comparaison sont utilisés avec la commande de requête `-find`.

**Valeur :** l'opérateur que vous souhaitez utiliser. Les opérateurs valides sont les suivants :

Mot-clé	Opérateur FileMaker Pro équivalent
<code>eq</code>	<code>=word</code>
<code>cn</code>	<code>*word*</code>
<code>bw</code>	<code>word*</code>
<code>ew</code>	<code>*word</code>
<code>gt</code>	<code>&gt; word</code>
<code>gte</code>	<code>&gt;= word</code>
<code>lt</code>	<code>&lt; word</code>
<code>lte</code>	<code>&lt;= word</code>
<code>neq</code>	<code>omit, word</code>

**Facultatif pour :** `-find`, commande de requête

**Nécessite :** Un nom de rubrique et une valeur.

La syntaxe de spécification d'un opérateur de comparaison est la suivante :

```
table-name::field-name=value&table-name::field-name.op=op-symbol
```

où :

- `table-name` est la table contenant la rubrique et est uniquement requise si la rubrique n'est pas dans la table source du modèle spécifié dans la chaîne de requête.
- `op-symbol` est l'un des mots-clés du tableau précédent, tel que `cn`.

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&name=Tim&name.op=cn&-find
```

**Remarque** Le mot-clé `bw` ne fonctionne pas avec les chaînes de date, d'heure ou d'horodatage, ou avec l'opérateur de recherche de date du jour (`//`).

Vous pouvez utiliser tout opérateur de recherche FileMaker Pro en l'incluant dans les critères de recherche au lieu de spécifier le mot-clé de l'opérateur `fieldname.op`. Par exemple, pour trouver une plage de valeurs à l'aide de l'opérateur de recherche de plage (`...`), ne spécifiez aucun mot-clé d'opérateur. Au lieu de cela, utilisez les caractères « `...` » entre les valeurs dans les critères de recherche.

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&IDnum=915...925&-find
```

Pour en savoir plus sur les opérateurs pouvant être utilisés pour rechercher du texte, consultez l'[Aide FileMaker Pro](#).

### Paramètre de requête `-lay` (modèle)

Indique le modèle de base de données que vous souhaitez utiliser.

**Valeur :** Nom du modèle.

**Obligatoire pour :** toutes les commandes de requête, à l'exception de `-dbnames`, `-layoutnames` et `-scriptnames`.

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-view
```

### Paramètre de requête `-lay.response` (changer de modèle pour la réponse)

Indique que FileMaker Server doit utiliser le modèle spécifié par le paramètre `-lay` lors du traitement d'une requête et passer au modèle spécifié par le paramètre `-lay.response` pour le traitement de la réponse XML.

Si vous n'incluez pas le paramètre `-lay.response`, FileMaker Server utilise le modèle spécifié par le paramètre `-lay` pour le traitement de la requête et de la réponse.

Vous pouvez utiliser le paramètre `-lay.response` pour les requêtes XML.

**Valeur :** Nom du modèle.

**Facultatif pour :** toutes les commandes de requête, à l'exception de `-dbnames`, `-layoutnames` et `-scriptnames`.

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=Budget&Salary=100000&Salary.op=gt&-find&-lay.response=ExecList
```

## Paramètre de requête `-lop` (opérateur logique)

Indique comment les critères de recherche de la commande de requête `-find` sont combinés en une recherche « and » ou « or ».

**Valeur :** `and` ou `or`

Si le paramètre de requête `-lop` n'est pas inclus, la commande de requête `-find` utilise la valeur « and ».

**Facultatif pour :** `-find`, commande de requête

**Remarque** Non pris en charge par la commande de requête `-findquery`.

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&Last+Name=Smith&Birthdate=2/5/1972&-lop=and&-find
```

## Paramètres de requête `-max` (nombre maximal d'enregistrements)

Indique le nombre maximum d'enregistrements que vous souhaitez obtenir.

**Valeur :** Un nombre ou la valeur `all` pour obtenir tous les enregistrements. Si `-max` n'est pas spécifié, tous les enregistrements sont renvoyés.

**Facultatif pour :** les commandes de requêtes `-find`, `-findall` et `-findquery`

**Remarque** Le paramètre de requête `-max` n'affecte pas les valeurs renvoyées pour les enregistrements externes. Pour limiter le nombre de lignes renvoyées pour les enregistrements externes, consultez « Paramètre de requête `-relatedsets.max` (limitation d'enregistrements de table externe) », page 59.

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-max=10&-findall
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-max=all&-findall
```

## Paramètre de requête `-modid` (ID de la modification)

L'ID de modification est un compteur incrémentiel indiquant la version actuelle d'un enregistrement. En spécifiant un ID de modification lorsque vous utilisez une commande de requête `-edit`, vous pouvez vous assurer que vous éditez la version actuelle d'un enregistrement. Si l'ID de modification que vous spécifiez ne correspond pas à la valeur d'ID de modification actuelle dans la base de données, la commande de requête `-edit` n'est pas autorisée et un code d'erreur est renvoyé.

**Valeur :** Un ID de modification unique, indiquant la version actuelle d'un enregistrement dans une base de données FileMaker Pro.

**Facultatif pour :** la commande de requête `-edit`

**Nécessite :** le paramètre `-recid`

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=22&-modid=6&last_name=Jones&-edit
```



## Paramètre de requête `-query` (requête de recherche composée)

Indique les noms de requête et les critères de recherche d'une requête de recherche composée. Consultez la section « Commande de requête `-findquery` (recherche composée) », page 50.

**Valeur** : Une expression de requête.

**Obligatoire pour** : `-findquery`, commande de requête

La syntaxe d'une requête de recherche composée est :

```
-query=<request-declarations><request-definitions>&-findquery
```

où :

`<request-declarations>` représente au moins deux déclarations de requête.

- Chaque déclaration de requête se compose d'un ou plusieurs identificateur(s) de requête, séparé(s) par des virgules et mis entre parenthèses. Un identificateur de requête est représenté par la lettre « q » suivie d'un nombre. Par exemple : `q1`
- Entourées de parenthèses, les requêtes multiples agissent comme un opérateur logique ET, ce qui réduit le résultat de la recherche. Par exemple, `(q1, q2)` renvoie des enregistrements correspondant à `q1` et `q2`.

**Remarque** Il est déconseillé d'utiliser les mêmes rubriques pour les variables `q` dans le même critère de recherche « and ».

- Comme avec FileMaker Pro, chaque requête peut être une recherche ou une exclusion. Une requête de recherche ajoute les enregistrements correspondants au jeu d'enregistrements trouvés ; une exclusion les en retire. Par défaut, les requêtes sont des recherches. Pour une requête d'exclusion, placer un point d'exclamation ( ! ) devant la parenthèse ouvrante.

### Exemple

```
(q1) ; ! (q2)
```

`q1` est une requête de recherche ; `q2` est une requête d'exclusion, car elle est précédée d'un point d'exclamation.

- Les requêtes sont séparées par des points-virgules. Les requêtes multiples agissent comme un opérateur logique OU, ce qui élargit le résultat de la recherche. Par exemple, `(q1) ; (q2)` renvoie des enregistrements correspondant à `q1` ou `q2`. Les requêtes d'exclusion n'agissent pas comme un opérateur logique OU car elles excluent des enregistrements du résultat de la recherche.
- Les requêtes sont exécutées dans l'ordre indiqué ; le jeu d'enregistrements trouvé inclut les résultats de la totalité de la requête de recherche composée.

`<request-definitions>` est une définition de requête correspondant à chaque déclaration de requête. Chaque définition de requête se compose d'une rubrique de recherche et d'une définition de valeur. Un signe moins (-) indique le début de la définition de requête.

Syntaxe :

```
-<query-id>=<fieldname>&-<query-id>.value=<value>
```

## Exemples

```
-q1=typeofanimal&-q1.value=Cat
-q2=name&-q2.value=Fluffy
```

Rechercher les enregistrements des chats gris qui ne s'appellent pas « Fluffy » :

```
http://host/fmi/xml/fmresultset.xml?-db=petclinic&-lay=Patients
&-query=(q1, q2);!(q3)&-q1=typeofanimal&-q1.value=Cat&-q2=color
&-q2.value=Gray&-q3=name&-q3.value=Fluffy&-findquery
```

## Paramètre de requête `-recid` (ID d'enregistrement)

Indique l'enregistrement que vous souhaitez traiter. Utilisé principalement dans les commandes de requête `-edit` et `-delete`. Utilisé par la commande `-view` pour extraire des données de liste de valeurs liées dans la grammaire FMPXMLLAYOUT.

**Valeur :** Un ID d'enregistrement qui est l'identifiant unique d'un enregistrement dans une base de données FileMaker Pro.

**Obligatoire pour :** les commandes de requête `-edit`, `-delete` et `-dup`

**Facultatif pour :** les commandes `-find` et `-view`

## Exemples

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=22&-delete
http://localhost/fmi/xml/FMPXMLLAYOUT.xml?-db=test&-lay=empty&-view
&-recid=9
```

## Paramètre de requête `-relatedsets.filter` (filtrage d'enregistrements de table externe)

Spécifie si les enregistrements de table externe à renvoyer dans les résultats de cette requête doivent être limités.

**Valeur :** `layout` ou `none`

- Si `-relatedsets.filter` est paramétré sur `layout`, le paramètre **Rangée initiale** spécifié dans la boîte de dialogue Table externe de FileMaker Pro est respecté.
  - Si le paramètre **Autoriser le défilement vertical** est activé dans la boîte de dialogue Table externe, utilisez l'option `-relatedsets.max` pour indiquer le nombre maximum d'enregistrements à renvoyer. Consultez la rubrique « Paramètre de requête `-relatedsets.max` (limitation d'enregistrements de table externe) » ci-dessous.
  - Si le paramètre **Autoriser le défilement vertical** est désactivé ou que l'option `-relatedsets.max` n'est pas utilisée, le paramètre **Nombre de rangées** de la boîte de dialogue Table externe détermine le nombre maximal d'enregistrements à renvoyer.
- La valeur par défaut est `none` si ce paramètre n'est pas précisé. Si `-relatedsets.filter` est paramétré sur `none`, le moteur de publication Web renvoie tous les enregistrements de la table externe. Les valeurs **Rangée initiale** et **Nombre de rangées** spécifiées dans la boîte de dialogue Table externe sont ignorées.

## Remarques

- Le paramètre `-relatedsets.filter` n'a aucun impact sur le mode de tri des enregistrements dans les requêtes XML. Le tri spécifié dans FileMaker Pro est respecté si la valeur de paramètre `-relatedsets.filter` est `layout` ou `none`.
- Le paramètre **Filtrer les enregistrements de la table externe** dans la boîte de dialogue Table externe n'est pas compatible avec les requêtes XML. Tout calcul spécifié pour le paramètre **Filtrer les enregistrements de la table externe** est ignoré.

**Facultatif pour :** `-find`, `-edit`, `-new`, `-dup` et `-findquery`.

## Exemples

```
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English
&-relatedsets.filter=none&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample
&-lay=English&relatedsets.filter=layout&-relatedsets.max=all&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English
&-relatedsets.filter=layout&-relatedsets.max=10&-findany
```

## Paramètre de requête `-relatedsets.max` (limitation d'enregistrements de table externe)

Indique le nombre maximum d'enregistrements externes à renvoyer dans les résultats de cette requête.

**La valeur est :** un nombre entier ou `all`.

- Le paramètre `-relatedsets.max` est respecté seulement si le paramètre **Autoriser le défilement vertical** est activé dans la boîte de dialogue Table externe de FileMaker Pro et si le paramètre `-relatedsets.filter` est défini sur `layout`.
  - Si le paramètre `-relatedsets.max` indique un nombre entier, le moteur de publication Web renvoie ce nombre d'enregistrements externes après la rangée initiale.
  - Si le paramètre `-relatedsets.max` indique `all`, alors le moteur de publication Web renvoie tous les enregistrements de la table externe.

**Remarque** Pour obtenir des informations sur le filtrage des enregistrements externes, consultez la rubrique « Paramètre de requête `-relatedsets.filter` (filtrage d'enregistrements de table externe) » ci-dessus.

**Facultatif pour :** `-find`, `-edit`, `-new`, `-dup` et `-findquery`.

## Exemples

```
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample
&-lay=English&relatedsets.filter=layout&-relatedsets.max=all&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English
&-relatedsets.filter=layout&-relatedsets.max=10&-findany
```

### Paramètre de requête `-script` (Script)

Indique le script FileMaker à lancer quand la commande de requête et le tri ont été exécutés. Consultez la section « Comprendre le traitement d'une requête XML », page 42.

**Valeur :** Nom du script

**Facultatif pour :** toutes les commandes de requête, à l'exception de `-dbnames`, `-layoutnames` et `-scriptnames`.

#### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script=myscript&-findall
```

### Paramètre de requête `-script.param` (transfert du paramètre au script)

Transmet un paramètre au script FileMaker spécifié par `-script`

**Valeur :** Un simple paramètre texte.

- Pour transmettre plusieurs paramètres, vous pouvez créer une chaîne délimitant les paramètres et faire en sorte que votre script analyse chacun des paramètres. Par exemple, transmettez `param1|param2|param3` en tant que liste en utilisant le caractère `|` de codage URL de la façon suivante : `param1%7Cparam2%7Cparam3`
- Pour traiter le paramètre texte en tant que valeur qui ne soit pas du texte, votre script peut convertir la valeur texte. Par exemple, pour convertir la valeur texte en nombre, votre script doit intégrer : `ObtenirNombre(Obtenir ( ParamètreScript ))`
- Si votre requête contient `-script.param` sans `-script`, alors `-script.param` est ignoré.
- Si votre requête contient plusieurs `-script.param`, alors le moteur de publication Web utilise la dernière valeur analysée.

**Facultatif pour :** `-script`

#### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script=myscript&-script.param=Smith%7CChatterjee%7CSu
&-findall
```

### Paramètre de requête `-script.prefind` (script avant une recherche)

Spécifie le script FileMaker à exécuter avant la recherche et le tri d'enregistrements (le cas échéant), pendant le traitement d'une commande de requête `-find`.

**Valeur :** Nom du script

**Facultatif pour :** toutes les commandes de requête, à l'exception de `-dbnames`, `-layoutnames` et `-scriptnames`.

#### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.prefind=myscript&-findall
```

### Paramètre de requête `-script.prefind.param` (transfert du paramètre au script avant recherche)

Transfère un paramètre au script FileMaker spécifié par `-script.prefind`

**Valeur :** Un simple paramètre texte.

- Pour transmettre plusieurs paramètres, vous pouvez créer une chaîne délimitant les paramètres et faire en sorte que votre script analyse chacun des paramètres. Par exemple, transmettez `param1|param2|param3` en tant que liste en utilisant le caractère `|` de codage URL de la façon suivante : `param1%7Cparam2%7Cparam3`
- Pour traiter le paramètre texte en tant que valeur qui ne soit pas du texte, votre script peut convertir la valeur texte. Par exemple, pour convertir la valeur texte en nombre, votre script doit intégrer : `ObtenirNombre(Obtenir ( ParamètreScript ))`
- Si votre requête contient `-script.prefind.param` sans `-script.prefind`, alors `-script.prefind.param` est ignoré.
- Si votre requête contient plusieurs `-script.prefind.param`, alors le moteur de publication Web utilise la dernière valeur analysée.

**Facultatif pour :** `-script.prefind`

#### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.prefind=myscript&-script.prefind.param=payroll
&-findall
```

### Paramètre de requête `-script.presort` (script avant un tri)

Spécifie le script FileMaker à exécuter après l'extraction d'enregistrements (le cas échéant) et avant leur tri, pendant le traitement d'une commande de requête `-find`.

**Facultatif pour :** toutes les commandes de requête, à l'exception de `-dbnames`, `-layoutnames` et `-scriptnames`.

#### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.presort=myscript&-sortfield.1=dept
&-sortfield.2=rating&-findall
```

### Paramètre de requête `-script.presort.param` (transfert du paramètre au script avant recherche)

Transfère un paramètre au script FileMaker spécifié par `-script.presort`.

**Valeur :** Un simple paramètre texte.

- Pour transmettre plusieurs paramètres, vous pouvez créer une chaîne délimitant les paramètres et faire en sorte que votre script analyse chacun des paramètres. Par exemple, transmettez `param1|param2|param3` en tant que liste en utilisant le caractère `|` de codage URL de la façon suivante : `param1%7Cparam2%7Cparam3`

- Pour traiter le paramètre texte en tant que valeur qui ne soit pas du texte, votre script peut convertir la valeur texte. Par exemple, pour convertir la valeur texte en nombre, votre script doit intégrer : `ObtenirNombre(Obtenir ( ParamètreScript ))`
- Si votre requête contient `-script.presort.param` sans `-script.presort`, alors `-script.presort.param` est ignoré.
- Si votre requête contient plusieurs `-script.presort.param`, alors le moteur de publication Web utilise la dernière valeur analysée.

**Facultatif pour :** `-script.presort`

### Exemple

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.presort=myscript&-script.presort.param=18%7C65
&-sortfield.1=dept&-sortfield.2=rating&-findall
```

### Paramètre de requête `-skip` (enregistrements ignorés)

Indique le nombre d'enregistrements à ignorer dans le jeu d'enregistrements trouvés.

**Valeur :** Un nombre. Si la valeur est supérieure au nombre d'enregistrements contenus dans l'ensemble d'enregistrements trouvés, aucun enregistrement n'est affiché. La valeur par défaut est 0.

**Facultatif pour :** `-find`, commande de requête

### Exemple

Les 10 premiers enregistrements de l'ensemble d'enregistrements trouvés sont ignorés et les enregistrements 11 à 15 sont affichés :

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-skip=10&-max=5&-findall
```

### Paramètre de requête `-sortfield` (rubrique de tri)

Spécifie la rubrique à utiliser pour le tri.

**La valeur est :** field name

**Facultatif pour :** les commandes de requête `-find` ou `-findall`

Le paramètre de requête `-sortfield` peut être utilisé plusieurs fois pour effectuer des tris de rubriques multiples. La syntaxe permettant de spécifier la priorité des rubriques de tri est la suivante :

```
-sortfield.precedence-number=fully-qualified-field-name
```

où `precedence-number` dans le paramètre de requête `-sortfield.precedence-number` est un nombre qui indique la priorité à utiliser dans le cas de rubriques à tri multiple. Valeur de `precedence-number` :

- doit commencer par 1.
- doit s'incrémenter de façon séquentielle.
- ne doit pas être supérieure à 9.

## Exemple

La rubrique « dept » est triée en premier, puis vient le tour de la rubrique « rating ». Les deux rubriques sont triées par ordre croissant car le paramètre de requête `-sortorder` n'est pas spécifié.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=performance&-sortfield.1=dept&-sortfield.2=rating&-findall
```

## Paramètre de requête `-sortorder` (ordre de tri)

Indique l'ordre d'un tri.

**Valeur :** L'ordre de tri. Les ordres de tri suivants sont valides, quand `<value-list-name>` est un nom de liste de valeurs tel que `Custom` :

Mot-clé	Opérateur FileMaker Pro équivalent
ascend	Tri a à z, -10 à 10
descend	Tri z à a, 10 à -10
<code>&lt;value-list-name&gt;</code>	Tri à l'aide de la liste des valeurs spécifiée, associée à la rubrique dans le modèle

**Facultatif pour :** les commandes de requête `-find` ou `-findall`

**Nécessite :** le paramètre de requête `-sortfield`

Le paramètre de requête `-sortorder` peut être utilisé avec le paramètre de requête `-sortfield` pour spécifier l'ordre de tri des rubriques de tri multiple. La syntaxe permettant de spécifier l'ordre de tri d'une rubrique de tri est la suivante :

```
-sortorder.precedence-number=sort-method
```

où :

- `precedence-number` dans le paramètre `-sortorder.precedence-number` est un nombre situé entre 1 et 9 indiquant le paramètre de requête `-sortfield` auquel le paramètre de requête `-sortorder` s'applique.
- `sort-method` est un des mots-clés du tableau précédent permettant de spécifier l'ordre de tri, par exemple `ascend`.

## Exemple

L'ordre de tri de la rubrique ayant la priorité la plus élevée (`dept`) est `ascend`, tandis que l'ordre de tri de la rubrique de priorité secondaire (`rating`) est `descend`. La valeur 2 affectée à `precedence-number` dans `-sortorder.2` indique que le paramètre de requête `-sortorder.2=descend` s'applique au paramètre de requête `-sortfield.2=rating`.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=performance&-sortfield.1=dept&-sortorder.1=ascend
&-sortfield.2=rating&-sortorder.2=descend&-findall
```

**Remarque** Si un paramètre de requête `-sortorder` n'est pas spécifié pour une rubrique de tri, le tri ascendant est utilisé par défaut.

# Chapitre 6

## À propos de la fonction de Publication Web personnalisée utilisant le langage PHP

La Publication Web personnalisée avec PHP vous permet d'utiliser le langage de script PHP pour intégrer des données provenant de bases de données FileMaker Pro dans vos modèles de pages Web personnalisés. La Publication Web personnalisée avec PHP fournit l'API FileMaker pour PHP, qui est une classe PHP qui accède aux bases de données hébergées par FileMaker Server. Cette classe PHP se connecte au moteur de publication Web de FileMaker Server et rend les données disponibles pour le moteur PHP du serveur Web.

### Fonctions clés de la Publication Web personnalisée avec PHP

- Créez des applications Web qui utilisent le langage de script PHP Open Source. Utilisez la version de PHP prise en charge par FileMaker Server ou utilisez votre propre version de PHP (si vous décidez d'utiliser votre propre version de PHP, consultez la section « Installation manuelle de l'API FileMaker pour PHP », page 66.).
- Hébergez des bases de données sur FileMaker Server. FileMaker Pro n'est pas requis pour la Publication Web personnalisée, car FileMaker Server héberge les bases de données.
- Rédigez un code PHP permettant de créer, de supprimer, d'éditer et de dupliquer les enregistrements d'une base de données FileMaker Pro hébergée. Votre code peut procéder à la validation des rubriques et des enregistrements avant de valider les modifications dans la base de données hébergée.
- Rédigez un code PHP permettant d'accéder aux modèles, aux tables externes, aux listes de valeurs et aux rubriques liées. Comme dans FileMaker Pro, l'accès aux données, aux modèles et aux rubriques est basé sur les paramètres de compte d'utilisateur définis dans les privilèges d'accès de la base de données. Le moteur de publication Web présente également plusieurs autres améliorations en matière de sécurité. Consultez la section « Protection de vos bases de données publiées », page 15.
- Rédigez un code PHP permettant d'exécuter des scripts complexes comportant plusieurs actions. La plateforme FileMaker prend en charge de nombreuses actions de script dans la Publication Web personnalisée. Consultez la section « Scripts FileMaker et Publication Web personnalisée », page 18.
- Rédigez un code PHP permettant d'effectuer des requêtes complexes.

### Configuration de la Publication Web personnalisée

Cette section décrit la configuration requise pour développer une solution de Publication Web personnalisée à l'aide de PHP, ce dont les utilisateurs Web ont besoin pour accéder à une solution de Publication Web personnalisée, ainsi que l'impact possible de l'hébergement d'une solution de publication Web sur votre serveur.



## Configuration requise pour la publication d'une base de données à l'aide de la Publication Web personnalisée

Pour publier des bases de données à l'aide de la fonction de Publication Web personnalisée avec PHP, il vous faut :

- un déploiement FileMaker Server, comprenant trois composants :
  - un serveur Web : Microsoft IIS (Windows) ou Apache (macOS) ; le module de serveur Web FileMaker est installé sur le serveur Web ;
  - le moteur de publication Web FileMaker ;
  - le serveur de bases de données FileMaker.
- une ou plusieurs bases de données FileMaker Pro hébergées par FileMaker Server ;
- l'adresse IP ou le nom de domaine de l'hôte exécutant le serveur Web ;
- un navigateur Web et un accès au serveur Web pour développer et tester la solution de publication Web personnalisée.
- une installation de PHP sur le serveur Web (FileMaker Server peut installer la version PHP prise en charge ou vous pouvez utiliser votre propre version) :
  - Pour connaître la version minimale requise de PHP, reportez-vous aux [spécifications techniques de FileMaker Server](#).
  - Pour plus d'informations sur PHP, consultez le site [php.net](http://php.net).
  - La version de PHP installée sur le serveur Web doit prendre en charge les fonctions cURL (bibliothèque d'URL du client). Pour plus d'informations sur cURL, consultez le site [php.net/curl](http://php.net/curl).

**Important** Quand vous installez FileMaker Server, la version de PHP prise en charge n'apparaît pas dans l'outil macOS Server Admin. Elle n'est pas supposée figurer dans la liste. Si vous utilisez l'outil macOS Server Admin pour activer PHP, vous désactivez la version de PHP prise en charge par FileMaker Server et vous activez votre propre version de PHP.

Consultez le [Guide d'installation et de configuration de FileMaker Server](#).

## Éléments devant être employés par les utilisateurs Web pour accéder à une solution de publication Web personnalisée

Pour accéder à une solution de Publication Web personnalisée utilisant PHP, les utilisateurs Web doivent posséder :

- un navigateur Web ;
- un accès à Internet ou à un intranet, ainsi qu'un accès au serveur Web ;
- l'adresse IP ou le nom de domaine de l'hôte exécutant le serveur Web.

Si la base de données est protégée par un mot de passe, les utilisateurs Web doivent également saisir un nom d'utilisateur et un mot de passe pour accéder au compte de la base de données.

## Connexion à Internet ou à un intranet

Lorsque vous publiez des bases de données sur Internet ou sur un intranet, l'ordinateur hôte doit exécuter FileMaker Server et les bases de données que vous souhaitez partager doivent être hébergées et accessibles. En outre :

- Publiez votre base de données sur un ordinateur possédant une connexion permanente à Internet ou à un intranet. En effet, si la connexion n'est pas permanente, les utilisateurs Web doivent attendre que l'hôte se connecte à Internet ou à un intranet pour accéder à la base de données.
- L'ordinateur hôte du serveur Web, partie intégrante du déploiement FileMaker Server, doit posséder une adresse IP *statique* (permanente) dédiée ou un nom de domaine. Si vous vous connectez à Internet avec un fournisseur d'accès à Internet (FAI), votre adresse IP est peut-être *allouée de manière dynamique* (c'est-à-dire différente à chaque fois que vous vous connectez). Le cas échéant, les utilisateurs Web auront plus de difficultés à localiser une base de données. Si vous ne savez pas de quel type d'accès vous disposez, contactez votre fournisseur d'accès à Internet ou l'administrateur du réseau.

## Installation manuelle de l'API FileMaker pour PHP

Lorsque vous installez FileMaker Server, la version de PHP prise en charge par FileMaker est installée. Si un autre moteur PHP est déjà installé et configuré et que vous souhaitez uniquement ajouter l'API FileMaker pour PHP, installez manuellement la classe API FileMaker pour PHP pour qu'elle soit disponible dans vos scripts PHP. Utilisez l'interface de ligne de commande (CLI) pour désactiver la version de PHP prise en charge par FileMaker. Consultez l'Aide de l'interface de ligne de commande (CLI).

Si vous utilisez votre propre moteur PHP, veillez à effectuer les tâches de configuration suivantes dans votre version du moteur PHP :

- Activez le module cURL dans `php.ini`.
- Indiquez l'emplacement de l'API FileMaker pour PHP dans la variable `include_path` de `php.ini`.
- Si vous accédez à des bases de données contenant des dates et des heures, installez le [package Pear de date](#).

**Remarque** Pour connaître la version minimale requise de PHP, reportez-vous aux [spécifications techniques de FileMaker Server](#). Pour des résultats optimaux, utilisez la version de PHP appropriée.

### Pour que l'API FileMaker pour PHP soit accessible dans vos scripts PHP

Lorsque vous avez installé FileMaker Server, l'API FileMaker pour PHP a été incluse sous la forme d'un fichier `.zip` à l'emplacement suivant :

- Pour IIS (Windows) :  
`[disque]:\Program Files\FileMaker\FileMaker Server\Web Publishing\FM_API_for_PHP_Standalone.zip`  
où `[disque]` correspond au disque sur lequel réside le composant Serveur Web de votre déploiement FileMaker Server.
- Pour Apache (macOS) :  
`/Library/FileMaker Server/Web Publishing/FM_API_for_PHP_Standalone.zip`

Le fichier FM\_API\_for\_PHP\_Standalone.zip contient un fichier nommé FileMaker.php et un dossier nommé FileMaker. Décompressez ce fichier et copiez le fichier FileMaker.php ainsi que le dossier FileMaker dans l'un des emplacements suivants :

- le dossier dans lequel résident vos scripts PHP ;
  - Pour IIS (Windows) via HTTP ou HTTPS :  
`[disque]:\Program Files\FileMaker\FileMaker Server\HTTPServer\Conf`  
où [disque] correspond au lecteur sur lequel réside le composant Moteur de publication Web de votre déploiement FileMaker Server.
  - Pour Apache (macOS) via HTTP :  
`/Library/FileMaker Server/HTTPServer/htdocs`
- Pour Apache (macOS) via HTTPS :  
`/Library/FileMaker Server/HTTPServer/htdocs/httpsRoot`

**Remarque** Lorsque vous importez un certificat SSL personnalisé, les connexions de clients au serveur de base de données utilisent SSL et les connexions HTTP sont acheminées vers HTTPS. Avec un certificat SSL personnalisé sur votre serveur, utilisez le répertoire HTTPS pour l'hébergement des fichiers PHP de votre site.

- l'un des répertoires `include_path` de votre installation de PHP. L'emplacement par défaut pour macOS est `/usr/lib/php`.

## Étape suivante

- Utilisez l'interface de ligne de commande (CLI) pour activer la Publication Web personnalisée. Consultez l'[Aide FileMaker Server](#).
- Dans FileMaker Pro, ouvrez chaque base de données FileMaker Pro à publier et assurez-vous que chacune d'entre elles dispose des privilèges d'accès étendus pour la Publication Web personnalisée. Consultez la section « Activation de la Publication Web personnalisée dans une base de données », page 13.
- Pour savoir comment accéder aux données des bases de données FileMaker Pro à l'aide de l'API FileMaker pour PHP, consultez le chapitre 8, « Utilisation de l'API FileMaker pour PHP ».

# Chapitre 7

## Présentation de la Publication Web personnalisée avec PHP

L'API FileMaker pour PHP permet d'intégrer des données à partir de bases de données FileMaker Pro dans des solutions PHP. Ce chapitre décrit le fonctionnement de PHP avec le moteur de publication Web personnalisé de FileMaker Server. Pour plus d'informations sur l'API FileMaker pour PHP, consultez le chapitre 8, « Utilisation de l'API FileMaker pour PHP ».

### Fonctionnement du moteur de publication Web avec les solutions PHP

FileMaker Server comprend trois composants : un serveur Web, le moteur de publication Web et le serveur de bases de données. Consultez le [Guide de démarrage de FileMaker Server](#). Pour prendre en charge les solutions PHP, un moteur PHP est installé avec le serveur Web sur l'ordinateur maître. FileMaker Server héberge une solution PHP quand vous placez les fichiers PHP sur le serveur Web de l'ordinateur maître.

- Lorsqu'un utilisateur Web ouvre une solution PHP, le serveur Web achemine la demande vers le moteur PHP, qui traite le code PHP.
- Si le code PHP contient des appels vers l'API FileMaker pour PHP, ces appels sont interprétés et envoyés sous la forme de requêtes vers le moteur de publication Web.
- Le moteur de publication Web recherche les données dans les bases de données hébergées par le serveur de bases de données.
- Le serveur de bases de données envoie les données demandées au moteur de publication Web.
- Le moteur de publication Web envoie les données au moteur PHP sur le serveur Web en réponse à l'appel d'API.
- La solution PHP traite les données et les affiche pour l'utilisateur Web.

### Étapes générales de la Publication Web personnalisée avec PHP

1. Utilisez FileMaker Pro pour activer la Publication Web personnalisée pour une base de données. Consultez la section chapitre 2, « Préparation des bases de données pour la Publication Web personnalisée ».

**Remarque** Lors de la création de solutions PHP destinées à un utilisateur final, prenez soin d'utiliser des jeux de privilèges d'accès équivalents dans la base de données FileMaker Pro. Dans le cas contraire, vous pouvez avoir accès à des modèles et des fonctions de la base de données FileMaker Pro qui ne seront pas accessibles à l'utilisateur final, ce qui risque d'entraîner des problèmes de fonctionnement.

2. Utilisez l'interface de ligne de commande (CLI) de FileMaker Server pour activer la Publication Web personnalisée avec PHP. Consultez l'[Aide FileMaker Server](#).

3. Dans l'Admin Console, confirmez que le privilège étendu fmphp est activé pour chaque base de données FileMaker Pro publiée.
    - Dans l'Admin Console, cliquez sur la page **Bases de données**.
    - Sélectionnez **Afficher les privilèges étendus** dans le sélecteur de menu à côté de **Toutes les bases de données**. Le texte **FMPHP** s'affiche pour les bases de données dont le privilège étendu fmphp est activé.
  4. Utilisez les outils de création PHP pour créer votre solution PHP, en intégrant les fonctions de l'API FileMaker à votre code PHP pour accéder à vos données FileMaker. Consultez chapitre 8, « Utilisation de l'API FileMaker pour PHP ».
  5. Copiez ou déplacez les fichiers et la structure de répertoire de votre site vers le dossier suivant, sur le serveur Web de l'ordinateur maître.
    - Pour IIS (Windows) via HTTP ou HTTPS :  
`[disque]:\Program Files\FileMaker\FileMaker Server\HTTPServer\Conf`  
où [disque] est le disque de l'ordinateur maître de votre déploiement FileMaker Server.
    - Pour Apache (macOS) via HTTP :  
`/Library/FileMaker Server/HTTPServer/htdocs`
    - Pour Apache (macOS) via HTTPS :  
`/Library/FileMaker Server/HTTPServer/htdocs/httpsRoot`
- Remarque** Lorsque vous importez un certificat SSL personnalisé, les connexions de clients au serveur de base de données utilisent SSL et les connexions HTTP sont acheminées vers HTTPS. Avec un certificat SSL personnalisé, utilisez le répertoire HTTPS pour l'hébergement des fichiers PHP du site.
6. Si une rubrique de base de données Conteneur stocke une référence à un fichier et non le fichier lui-même, l'objet Conteneur référencé doit alors être stocké dans le dossier Web de FileMaker Pro lorsque l'enregistrement est créé ou modifié. Vous devez copier ou déplacer l'objet dans un dossier possédant le même emplacement relatif dans le dossier racine du logiciel de serveur Web.  
Consultez la section « À propos de la publication du contenu des rubriques Conteneur sur le Web », page 16.
  7. Assurez-vous que les mécanismes de sécurité pour votre site ou votre programme sont mis en œuvre.
  8. Testez votre site en utilisant les comptes et les privilèges d'accès définis pour les utilisateurs Web.
  9. Mettez le site à la disposition des utilisateurs. L'URL saisie par l'utilisateur Web utilise le format suivant :  
`http://<serveur>/<chemin_site>`
    - <serveur> est l'ordinateur sur lequel se trouve FileMaker Server.
    - <chemin\_site> est le chemin d'accès relatif vers la page d'accueil de votre site, déterminé par la structure de répertoire que vous avez utilisée à l'étape 5 ci-dessus.

### Exemple

Si l'adresse de votre serveur Web est 192.168.123.101 et que la page d'accueil de votre site se trouve sur le serveur Web à l'emplacement `c:\inetpub\wwwroot\customers\index.php`, l'utilisateur Web doit saisir l'URL suivante :

```
http://192.168.123.101/customers/index.php
```

**Remarque** PHP utilise le codage Latin-1 (ISO-8859-1). FileMaker Server renvoie des données Unicode (UTF-8). Utilisez l'interface de ligne de commande (CLI) pour spécifier le codage des caractères par défaut. Consultez l'Aide de l'interface de ligne de commande (CLI). Pour les sites PHP, vous pouvez spécifier UTF-8 ou ISO-8859-1 ; UTF-8 est recommandé. Spécifiez le même paramètre pour l'attribut `charset` dans la section `<HEAD>` des fichiers PHP de votre site.

Pour plus d'informations sur le déploiement et l'utilisation d'une solution PHP, consultez le chapitre 9, « Stockage, test et suivi d'un site ».

# Chapitre 8

## Utilisation de l'API FileMaker pour PHP

L'API FileMaker pour PHP utilise une classe PHP, la classe FileMaker, qui fournit une interface orientée objet aux bases de données FileMaker Pro. L'API de FileMaker pour PHP permet l'accès aux données et logiques stockées dans une base de données FileMaker Pro et leur publication sur le Web, ainsi que leur exportation vers d'autres applications.

L'API FileMaker pour PHP permet au code PHP d'effectuer le même type d'opérations que celles déjà disponibles dans les bases de données FileMaker Pro :

- créer, supprimer, modifier et dupliquer des enregistrements ;
- exécuter des requêtes ;
- procéder à la validation des rubriques et des enregistrements ;
- utiliser des modèles ;
- exécuter des scripts FileMaker ;
- afficher les tables externes et les enregistrements liés ;
- utiliser les listes de valeurs.

Ce chapitre explique comment utiliser les objets et méthodes de classe FileMaker pour ajouter ces fonctions courantes à une solution PHP. Il ne traite pas de l'API FileMaker pour PHP dans son intégralité, mais présente les objets et méthodes clés.

### Où trouver des informations supplémentaires

Pour en savoir plus sur l'API FileMaker pour PHP, consultez les ressources suivantes.

Si vous disposez déjà d'un moteur PHP installé et configuré et que vous souhaitez uniquement ajouter l'API FileMaker pour PHP, consultez la section « Installation manuelle de l'API FileMaker pour PHP », page 66.

### Référence API FileMaker pour PHP

Si vous avez installé l'API FileMaker pour PHP, vous pouvez trouver des informations de référence sur le composant Serveur Web de votre déploiement FileMaker Server.

- Pour IIS (Windows) :  
`[disque]:\Program Files\FileMaker\FileMaker Server\Documentation\PHP API Documentation\index.html`  
où [disque] correspond au disque sur lequel réside le composant Serveur Web de votre déploiement FileMaker Server.
- Pour Apache (macOS) : `/Library/FileMaker Server/Documentation/PHP API Documentation/index.html`

### Prise en charge API FileMaker pour PHP

Vous pouvez trouver des informations supplémentaires concernant l'API FileMaker pour PHP sur la page d'[assistance FileMaker](#).

## Utilisation de la classe FileMaker

Pour utiliser la classe FileMaker dans votre solution PHP, ajoutez l'instruction suivante à votre code PHP :

```
require_once ('FileMaker.php');
```

### Objets de classe FileMaker

La classe FileMaker définit des objets de classe que vous pouvez utiliser pour extraire des données à partir des bases de données FileMaker Pro.

Objet de classe	Utilisez cet objet pour
Base de données FileMaker Pro	Définir les propriétés de la base de données Se connecter à une base de données FileMaker Pro Obtenir des informations sur l'API FileMaker pour PHP
Commande	Créer des commandes qui ajoutent, suppriment, dupliquent ou modifient des enregistrements, effectuent des requêtes et exécutent des scripts
Modèle	Manipuler des modèles de base de données
Enregistrement	Manipuler des données d'enregistrement
Rubrique	Manipuler des données de rubrique
Ensemble lié	Manipuler des enregistrements de table externe
Résultat	Traiter les enregistrements renvoyés par une requête
Erreur	Vérifier si une erreur s'est produite Traiter les erreurs éventuelles

### Objets de commande FileMaker

La classe FileMaker définit un objet de commande de base qui vous permet d'instancier une commande spécifique et de spécifier les paramètres de cette commande. Pour exécuter la commande, vous devez appeler la méthode `execute()`.

La classe FileMaker définit les commandes spécifiques suivantes :

- Ajouter, commande
- Recherche composée, commande
- Supprimer, commande
- Dupliquer, commande
- Edition, commande
- Commande Rechercher, commande Rechercher tout, commande Rechercher au moins
- Commande Requête, qui s'ajoute à une commande de recherche composée
- Commande Exécuter le script



**Important** Les valeurs que les commandes renvoient diffèrent, comme défini par la classe `FileMaker.php`. Par exemple, certaines commandes renvoient la valeur booléenne `TRUE` ou un objet `FileMaker_Error`. D'autres commandes renvoient un objet `FileMaker_Result` qui peut contenir un « jeu » complet d'enregistrements dans un modèle. Pour éviter de surcharger la mémoire de l'ordinateur, tâchez de connaître les valeurs que renvoient les commandes que vous utilisez. Pour en savoir plus sur les valeurs que chaque commande renvoie, consultez la section « Référence API FileMaker pour PHP », page 71.

Les tâches de base que la plupart des applications PHP exécutent sont décrites de la manière suivante :

- « Utilisation des enregistrements », page 74
- « Exécution de scripts FileMaker », page 76
- « Exécution de requêtes », page 83

### Décodage des données à utiliser dans l'API FileMaker

Si votre application PHP récupère des données d'un site Web, ces données sont peut-être codées sous forme d'URL. L'API FileMaker pour PHP a besoin de données décodées et non de chaînes codées sous forme d'URL. En règle générale, il vous faut appeler la fonction `urldecode()` au moment où vous récupérez des données dans votre application PHP.

#### Exemple

```
$user = urldecode($_GET['user']);  
$event = urldecode($_GET['event']);
```

**Remarque** Avec l'API FileMaker pour PHP, évitez d'utiliser des chaînes qui intègrent les caractères esperluette (&). Dans les chaînes transmises à l'API FileMaker pour PHP, utilisez une barre oblique inverse comme caractère d'échappement avant les caractères spéciaux.

## Connexion à une base de données FileMaker Pro

La classe `FileMaker` définit un objet de base de données que vous instanciez pour vous connecter à un serveur ou à une base de données. Définissez les propriétés de l'objet avec le constructeur de classe ou en appelant la méthode `setProperty()`.

## Exemples

Connexion à un serveur pour obtenir une liste des bases de données :

```
$fm = new FileMaker();  
$databases = $fm->listDatabases();
```

Connexion à une base de données spécifique sur un serveur :

```
$fm = new FileMaker();  
$fm->setProperty('database', 'questionnaire');  
$fm->setProperty('hostspec', 'http://192.168.100.110');  
$fm->setProperty('username', 'web');  
$fm->setProperty('password', 'web');
```

Les propriétés du nom d'utilisateur et du mot de passe déterminent le jeu de privilèges d'accès de cette connexion.

**Remarque** La propriété `hostspec` prend la valeur `http://localhost` par défaut. Dans la mesure où le moteur PHP est installé avec le composant de serveur Web de l'ordinateur maître, il est inutile de spécifier la propriété `hostspec`.

## Utilisation des enregistrements

La classe FileMaker définit un objet d'enregistrement que vous instanciez pour manipuler les enregistrements. Une instance d'objet d'enregistrement représente un enregistrement à partir d'une base de données FileMaker Pro. Utilisez un objet d'enregistrement avec les commandes Ajouter, Supprimer, Dupliquer et Edition pour modifier les données de l'enregistrement. Les commandes Rechercher (Rechercher, Rechercher tout, Rechercher au moins et Recherche composée) renvoient un tableau d'objets d'enregistrement.

### Création d'un enregistrement

Il existe deux manières de créer un enregistrement :

- Utilisez la méthode `createRecord()`, en spécifiant un nom de modèle et éventuellement un tableau de valeurs de rubriques. Vous pouvez également définir des valeurs individuellement dans le nouvel objet d'enregistrement.

La méthode `createRecord()` ne sauvegarde pas le nouvel enregistrement dans la base de données. Pour sauvegarder l'enregistrement dans la base de données, appelez la méthode `commit()`.

#### Exemple

```
$rec = $fm->createRecord('Form View', $values);  
$result = $rec->commit();
```

Avec la méthode FileMaker\_Record `commit()`, la variable `$result` se voit affecter la valeur booléenne TRUE et le nouvel enregistrement est créé dans la base de données FileMaker pro en l'absence d'erreur.

Si une erreur se produit, la variable `$result` contient un objet FileMaker\_Error. Recherchez les erreurs après avoir exécuté la méthode `commit()`.

- Utilisez la commande Ajouter. Utilisez la méthode `newAddCommand()` pour créer un objet `FileMaker_Command_Add`, en spécifiant le nom du modèle et un tableau de données d'enregistrement. Pour sauvegarder l'enregistrement dans la base de données, appelez la méthode `execute()`.

### Exemple

```
$newAdd = $fm->newAddCommand('Respondent', $respondent_data);  
$result = $newAdd->execute();
```

Avec la méthode `FileMaker_Command execute()`, la variable `$result` contient un objet `FileMaker_Result` et toutes les informations au sujet de l'enregistrement créé en l'absence d'erreur.

Si une erreur se produit, la variable `$result` contient un objet `FileMaker_Error`. Recherchez les erreurs après avoir exécuté la méthode `execute()`.

## Duplication d'un enregistrement

Dupliquez un enregistrement existant en utilisant la commande Dupliquer. Utilisez la méthode `newDuplicateCommand()` pour créer un objet `FileMaker_Command_Duplicate`, en spécifiant le nom du modèle et l'ID de l'enregistrement que vous souhaitez dupliquer. Puis, dupliquez l'enregistrement en appelant la méthode `execute()`.

### Exemple

```
$newDuplicate = $fm->newDuplicateCommand('Respondent', $rec_ID);  
$result = $newDuplicate->execute();
```

## Edition d'un enregistrement

Il existe deux manières d'éditer un enregistrement :

- Utilisation de la commande Edition. Utilisez la méthode `newEditCommand()` pour créer un objet `FileMaker_Command_Edit`, en spécifiant le nom du modèle et l'ID de l'enregistrement que vous souhaitez éditer, ainsi qu'un tableau de valeurs que vous souhaitez mettre à jour. Puis, éditez l'enregistrement en appelant la méthode `execute()`.

### Exemple

```
$newEdit = $fm->newEditCommand('Respondent', $rec_ID, $respondent_data);  
$result = $newEdit->execute();
```

- Utilisation d'un objet d'enregistrement. Extrayez un enregistrement à partir de la base de données, modifiez les valeurs des rubriques, puis éditez l'enregistrement en appelant la méthode `commit()`.

### Exemple

```
$rec = $fm->getRecordById('Form View', $rec_ID);
$rec->setField('Name', $nameEntered);
$result = $rec->commit();
```

## Suppression d'un enregistrement

Il existe deux manières de supprimer un enregistrement :

- Extrayez un enregistrement à partir de la base de données, puis appelez la méthode `delete()`.

### Exemple

```
$rec = $fm->getRecordById('Form View', $rec_ID);
$rec->delete();
```

- Supprimez un enregistrement existant en utilisant la commande Supprimer. Utilisez la méthode `newDeleteCommand()` pour créer un objet `FileMaker_Command_Delete`, en spécifiant le nom du modèle et l'ID de l'enregistrement que vous souhaitez supprimer. Puis, supprimez l'enregistrement en appelant la méthode `execute()`.

### Exemple

```
$newDelete = $fm->newDeleteCommand('Respondent', $rec_ID);
$result = $newDelete->execute();
```

## Exécution de scripts FileMaker

Un script FileMaker est un ensemble nommé d'actions de script. La classe FileMaker définit plusieurs méthodes qui vous permettent d'utiliser des scripts FileMaker définis dans une base de données FileMaker Pro. Pour plus d'informations sur les actions de script compatibles Web (les actions de script qui peuvent être exécutées dans une solution Web), consultez la section « Scripts FileMaker et Publication Web personnalisée », page 18.

### Obtention de la liste des scripts disponibles

Utilisez la méthode `listScripts()` pour obtenir la liste des scripts disponibles à partir de la base de données actuellement connectée. La méthode `listScripts()` renvoie un tableau de scripts pouvant être exécutés par le nom d'utilisateur et le mot de passe spécifiés lors de la définition de la connexion à la base de données. (Consultez la section « Connexion à une base de données FileMaker Pro », page 73.)

### Exemple

```
$scripts = $fm->listScripts();
```

## Exécution d'un script FileMaker

Utilisez la méthode `newPerformScriptCommand()` pour créer un objet `FileMaker_Command_PerformScript`, spécifiant le modèle, le nom du script et les paramètres de script de votre choix. Puis, exécutez le script en appelant la méthode `execute()`.

**Important** Quand vous exécutez un script FileMaker, la taille de l'objet `FileMaker_Result` renvoyé dépend du comportement du script FileMaker. Par exemple, si un script FileMaker bascule sur un modèle en particulier, tous les enregistrements de la table de ce modèle peuvent figurer dans le jeu trouvé, et tous les enregistrements du jeu trouvé peuvent être renvoyés dans l'objet `FileMaker_Result`. Pour éviter de surcharger la mémoire de l'ordinateur, tâchez d'estimer les données renvoyées par un script FileMaker avant de l'exécuter dans une application PHP.

### Exemple

```
$newPerformScript = $fm->newPerformScriptCommand('Order Summary',  
'ComputeTotal');  
$result = $newPerformScript->execute();
```

## Exécution d'un script avant l'exécution d'une commande

Utilisez la méthode `setPreCommandScript()` pour spécifier l'exécution d'un script avant l'exécution d'une commande. L'exemple suivant utilise une commande Rechercher, mais vous pouvez utiliser la méthode `setPreCommandScript()` avec n'importe quelle commande.

### Exemple

```
$findCommand = $fm->newFindCommand('Students');  
$findCommand->addFindCriterion('GPA', $searchValue);  
$findCommand->setPreCommandScript('UpdateGPA');  
$result = $findCommand->execute();
```

## Exécution d'un script avant le tri d'un ensemble de résultats

Utilisez la méthode `setPreSortScript()` pour spécifier l'exécution d'un script après la génération d'un ensemble de résultats à l'aide de la commande Rechercher, mais avant le tri de cet ensemble de résultats. Consultez la section « Utilisation de la commande Rechercher », page 85.

### Exemple

```
$findCommand = $fm->newFindCommand('Students');  
$findCommand->setPreSortScript('RemoveExpelled');
```

## Exécution d'un script après la génération d'un ensemble de résultats

Utilisez la commande `setScript()` pour spécifier l'exécution d'un script après la génération d'un ensemble de résultats à l'aide de la commande Rechercher. Consultez la section « Utilisation de la commande Rechercher », page 85.

## Exemple

```
$findCommand = $fm->newFindCommand('Students');  
$findCommand->setScript('myScript', 'param1|param2|param3');
```

## Ordre d'exécution du script

Vous pouvez spécifier les méthodes `setPreCommandScript()`, `setPreSortScript()` et `setScript()` en conjonction avec les méthodes `setResultLayout()` et `addSortRule()` pour une seule commande.

Voici l'ordre dans lequel FileMaker Server et le moteur de publication Web traitent ces méthodes :

1. Exécution du script spécifié dans la méthode `setPreCommandScript()`, le cas échéant.
2. Traitement de la commande elle-même, par exemple la commande Rechercher ou Supprimer l'enregistrement.
3. Exécution du script spécifié dans la méthode `setPreSortScript()`, le cas échéant.
4. Tri de l'ensemble de résultats de la commande Rechercher, si la méthode `addSortRule()` a été spécifiée.
5. Traitement de la méthode `setResultLayout()` pour passer d'un modèle à un autre, si indiqué.
6. Exécution du script spécifié dans la méthode `setScript()`, le cas échéant.
7. Renvoi de l'ensemble de résultats final de la commande Rechercher.

Si l'une des étapes décrites ci-dessus génère un code d'erreur, l'exécution de la commande s'arrête ; les étapes suivantes ne sont alors pas exécutées. Toutefois, toutes les étapes antérieures dans la requête restent exécutées.

Par exemple, imaginons une commande qui supprime l'enregistrement ouvert, trie les enregistrements, puis exécute un script. Si la méthode `addSortRule()` pointe vers une rubrique qui n'existe pas, la requête supprime l'enregistrement ouvert et renvoie le code d'erreur 102 (« Rubrique manquante »), mais n'exécute cependant pas le script.

Le modèle indiqué pour la méthode `newFindCommand()` est utilisé lors du traitement de la requête. Au moment où la méthode `setResultLayout()` bascule sur un autre modèle, l'objet d'erreur pour la requête d'après le modèle d'origine n'est plus disponible. Pour tester l'objet d'erreur depuis la requête, d'après le modèle d'origine, vérifiez celui-ci avant de modifier le modèle.

## Exemple

```

request = $fm->newFindCommand('Students');
$request->addFindCriterion('Day', 'Wednesday');

// Exécuter la requête
$result = $request->execute();

if (FileMaker::isError($result)) {
    if ($result->code = 401) {
        $findError = 'Aucun enregistrement ne correspond à la requête : ' . ' (' .
            $result->code . ')';
    } else {
        $findError = 'Trouver l'erreur : ' . $result->getMessage() . ' (' .
            $result->code . ')';
    }
}
$request->setResultLayout('Teachers');
// Basculez sur le modèle des résultats
$result = $request->execute();

```

## Utilisation des modèles FileMaker Pro

Un modèle est un agencement de rubriques, d'objets, d'images et d'autres éléments qui déterminent l'organisation et la présentation des informations lors de la lecture, de la prévisualisation ou de l'impression d'enregistrements par l'utilisateur. La classe FileMaker définit plusieurs méthodes qui vous permettent d'utiliser des modèles définis dans une base de données FileMaker Pro. Vous pouvez obtenir des informations sur les modèles à partir de plusieurs objets de classe FileMaker.

### Avec cet objet de classe Utilisez ces méthodes

Base de données	<ul style="list-style-type: none"> <li>▪ <code>listLayouts()</code> obtient la liste des noms de modèles disponibles.</li> <li>▪ <code>getLayout()</code> obtient un objet de modèle en pointant vers un nom de modèle.</li> </ul>
Modèle	<ul style="list-style-type: none"> <li>▪ <code>getName()</code> extrait le nom de modèle d'un objet de modèle spécifique.</li> <li>▪ <code>listFields()</code> extrait un tableau de tous les noms de rubriques utilisés dans un modèle.</li> <li>▪ <code>getFields()</code> extrait un tableau associatif contenant les noms de toutes les rubriques sous la forme de clés et les objets FileMaker_Field liés sous la forme d'un tableau de valeurs.</li> <li>▪ <code>listValueLists()</code> extrait un tableau des noms de listes de valeurs.</li> <li>▪ <code>listRelatedSets()</code> extrait un tableau des noms des ensembles liés.</li> <li>▪ <code>getDatabase()</code> renvoie le nom de la base de données.</li> </ul>
Enregistrement	<ul style="list-style-type: none"> <li>▪ <code>getLayout()</code> renvoie l'objet de modèle lié à un enregistrement spécifique.</li> </ul>
Rubrique	<ul style="list-style-type: none"> <li>▪ <code>getLayout()</code> renvoie l'objet de modèle contenant une rubrique spécifique.</li> </ul>
Commande	<ul style="list-style-type: none"> <li>▪ <code>setResultLayout()</code> renvoie les résultats de la commande dans un modèle différent du modèle en cours.</li> </ul>

## Utilisation des tables externes

Une table externe est une table qui affiche les lignes de données d'un ou plusieurs enregistrements correspondants. La classe FileMaker définit un objet d'ensemble lié et plusieurs méthodes qui vous permettent d'utiliser des tables externes définies dans une base de données FileMaker Pro.

Un objet d'ensemble lié est un tableau d'objets d'enregistrement issus de la table externe correspondante ; chaque objet d'enregistrement correspond à une ligne de données dans la table externe.

### Création d'une liste des tables externes définies sur un modèle spécifique

Pour un objet de modèle spécifique, utilisez la méthode `listRelatedSets()` pour extraire une liste des noms de tables de toutes les tables externes définies dans ce modèle.

#### Exemple

```
$tableNames = $currentLayout->listRelatedSets();
```

### Obtention des noms de tables externes d'un objet de résultat spécifique

Pour un objet FileMaker\_Result spécifique, utilisez la méthode `getRelatedSets()` pour extraire les noms de toutes les tables externes de cet enregistrement.

#### Exemple

```
$relatedSetsNames = $result->getRelatedSets();
```

### Obtention d'informations sur les tables externes d'un modèle spécifique

Pour un objet de modèle spécifique, utilisez la méthode `getRelatedSets()` pour extraire un tableau d'objets FileMaker\_RelatedSet décrivant les tables externes du modèle. Le tableau renvoyé est un tableau associatif contenant les noms des tables sous la forme d'un tableau de clés, et les objets FileMaker\_RelatedSet liés sous la forme d'un tableau de valeurs.

#### Exemple

```
$relatedSetsArray = $currentLayout->getRelatedSets();
```

### Obtention d'informations sur une table externe spécifique

Pour un objet de modèle spécifique, utilisez la méthode `getRelatedSet()` pour extraire l'objet FileMaker\_RelatedSet décrivant une table externe spécifique.

#### Exemple

```
$relatedSet = $currentLayout->getRelatedSet('customers');
```



## Obtention du nom de table d'une table externe

Pour un objet d'ensemble lié, utilisez la méthode `getName()` pour obtenir le nom de table de la table externe.

### Exemple

```
$tableName = $relatedSet->getName();
```

## Obtention des enregistrements externes d'un enregistrement spécifique

Pour un objet d'enregistrement spécifique, utilisez la méthode `getRelatedSet()` pour extraire un tableau des enregistrements externes liés à cet enregistrement.

### Exemple

```
$relatedRecordsArray = $currentRecord->getRelatedSet('customers');
```

## Création d'un nouvel enregistrement dans une table externe

Utilisez la méthode `newRelatedRecord()` pour créer un nouvel enregistrement dans l'ensemble lié spécifié, et validez cette modification dans la base de données en appelant la méthode `commit()`.

### Exemple

```
//créer une nouvelle ligne de table externe dans la table externe 'customer'
$new_row = $currentRecord->newRelatedRecord('customer');

//définir les valeurs des rubriques dans la nouvelle ligne de table externe
$new_row->setField('customer::name', $newName);
$new_row->setField('customer::company', $newCompany);

$result = $new_row->commit();
```

## Suppression d'un enregistrement d'une table externe

Utilisez la méthode `delete()` pour supprimer un enregistrement d'une table externe.

### Exemple

```
$relatedSet = $currentRecord->getRelatedSet('customers');
/* Exécuté sur chacune des lignes de la table externe */
foreach ($relatedSet as $nextRow) {
    $nameField = $nextRow->getField('customer::name')
    if ($nameField == $badName) {
        $result = $newRow->delete();
    }
}
```

## Utilisation des listes de valeurs

Une liste de valeurs est un ensemble de choix prédéfinis. La classe FileMaker définit plusieurs méthodes qui vous permettent d'utiliser des listes de valeurs définies dans une base de données FileMaker Pro.

### Obtention des noms de toutes les listes de valeurs d'un modèle spécifique

Pour un objet de modèle spécifique, utilisez la méthode `listValueLists()` pour extraire un tableau contenant les noms des listes de valeurs.

#### Exemple

```
$valueListNames = $currentLayout->listValueLists();
```

### Obtention d'un tableau de toutes les listes de valeurs d'un modèle spécifique

Pour un objet de modèle spécifique, utilisez la méthode `getValueListsTwoFields()` pour extraire un tableau contenant les valeurs de toutes les listes de valeurs. Le tableau renvoyé est un tableau associatif. Les clés du tableau correspondent aux noms des listes de valeurs et les valeurs du tableau sont des tableaux associatifs qui dressent la liste des noms affichés et des choix correspondants pour chaque liste de valeurs.

#### Exemple

```
$valueListsArray = $currentLayout->getValueListsTwoFields();
```

**Remarque** Bien que la méthode `getValueLists()` soit encore prise en charge dans l'API FileMaker pour PHP, elle n'est plus utilisée. Utilisez plutôt la méthode `getValueListsTwoFields()`.

### Obtention des valeurs pour une liste de valeurs nommée

Pour un objet de modèle spécifique, utilisez la méthode `getValueListTwoFields()` pour extraire un tableau des choix définis dans la liste de valeurs nommée. Le tableau renvoyé est un tableau associatif. Ses clés correspondent aux valeurs affichées dans la seconde rubrique de la liste de valeurs et ses valeurs, aux valeurs stockées associées de la première rubrique.

En fonction des options sélectionnées dans la boîte de dialogue Rubriques pour la liste de valeurs de la base de données FileMaker Pro, la méthode `getValueListTwoFields()` renvoie la valeur dans la première rubrique uniquement, la valeur dans la seconde rubrique uniquement ou les valeurs dans les deux rubriques d'une liste de valeurs, l'une étant la valeur stockée et l'autre, la valeur affichée.

- Si l'option **Afficher également les valeurs de la seconde rubrique** n'est pas sélectionnée, la méthode `getValueListTwoFields()` renvoie la valeur de la première rubrique de la liste de valeurs, correspondant à la fois à la valeur stockée et à la valeur affichée.
- Si les options **Afficher également les valeurs de la seconde rubrique** et **Afficher uniquement les valeurs de la seconde rubrique** sont toutes les deux sélectionnées, la méthode `getValueListTwoFields()` renvoie la valeur de la première rubrique sous forme de valeur stockée et la valeur de la seconde rubrique sous forme de valeur affichée.

- Si l'option **Afficher également les valeurs de la seconde rubrique** est sélectionnée et que l'option **Afficher uniquement les valeurs de la seconde rubrique** ne l'est pas, la méthode `getValueListTwoFields()` renvoie la valeur de la première rubrique sous forme de valeur stockée et les deux valeurs provenant de la première et de la seconde rubriques, sous forme de valeur affichée.

Utilisez un itérateur avec la méthode `getValueListTwoFields()` pour rechercher la valeur affichée et la valeur stockée.

### Exemple

```
$layout = $fm->getLayout('customers');
$valuearray = $layout->getValueListTwoFields("region", 4);
foreach ($valuearray as $displayValue => $value) {
    ....
}
```

### Remarques

- Bien que la méthode `getValueList()` soit encore prise en charge dans l'API FileMaker pour PHP, elle n'est plus utilisée. Utilisez plutôt la méthode `getValueListTwoFields()`.
- Lorsque vous utilisez la méthode `getValueListTwoFields()`, veillez à inclure une boucle `foreach` pour faire une boucle dans le tableau associatif. Par contre, n'utilisez pas de boucle `for` qui pourrait générer des résultats inattendus.

## Exécution de requêtes

La classe FileMaker définit quatre types d'objets de commande Rechercher :

- Commande Rechercher tout. Consultez la section « Utilisation de la commande Rechercher tout », page 84.
- Commande Rechercher au moins. Consultez la section « Utilisation de la commande Rechercher au moins », page 84.
- Commande Rechercher. Consultez la section « Utilisation de la commande Rechercher », page 85.
- Commande Recherche composée. Consultez la section « Utilisation d'une commande Recherche composée », page 85.

La classe FileMaker définit également plusieurs méthodes pouvant être utilisées pour les quatre types de commandes Rechercher :

- Utilisez la méthode `addSortRule()` pour ajouter une règle définissant la manière dont l'ensemble de résultats doit être trié. Utilisez la méthode `clearSortRules()` pour effacer toutes les règles de tri qui ont été définies.
- Utilisez la méthode `setLogicalOperator()` pour basculer entre les recherches AND logiques et les recherches OR logiques.

- Utilisez la méthode `setRange()` pour effectuer une requête visant à obtenir uniquement une partie de l'ensemble de résultats. Utilisez la méthode `getRange()` pour extraire la définition de la plage actuelle.

L'utilisation de la méthode `setRange()` peut améliorer la performance de votre solution en réduisant le nombre d'enregistrement renvoyés par la requête Rechercher. Par exemple, si une requête Rechercher renvoie 100 enregistrements, vous pouvez répartir l'ensemble de résultats en cinq groupes de 20 enregistrements chacun au lieu de traiter les 100 enregistrements en une seule fois.

- Vous pouvez exécuter les scripts FileMaker en conjonction avec les commandes Rechercher.
  - Pour exécuter un script avant d'exécuter la commande Rechercher, utilisez la méthode `setPreCommandScript()`.
  - Pour exécuter un script avant le tri de l'ensemble de résultats, utilisez la méthode `setPreSortScript()`.
  - Pour exécuter un script après la génération d'un ensemble de résultats, mais avant le tri de cet ensemble de résultats, utilisez la méthode `setScript()`.

### Utilisation de la commande Rechercher tout

Utilisez la commande Rechercher tout pour extraire tous les enregistrements d'un modèle spécifié. Utilisez la méthode `newFindAllCommand()`, en spécifiant un modèle spécifique, pour créer un objet `FileMaker_Command_FindAll`. Puis, exécutez la requête en appelant la méthode `execute()`.

#### Exemple

```
$findCommand = $fm->newFindAllCommand('Form View');  
$result = $findCommand->execute;
```

**Remarque** Pour éviter les problèmes de surcharge de la mémoire de l'ordinateur si vous utilisez la commande Rechercher tout, indiquez par défaut un nombre maximum d'enregistrements à renvoyer par page.

### Utilisation de la commande Rechercher au moins

Utilisez la commande Rechercher au moins pour extraire un enregistrement aléatoire d'un modèle spécifié. Utilisez la méthode `newFindAnyCommand()`, en spécifiant un modèle spécifique, pour créer un objet `FileMaker_Command_FindAny`. Puis, exécutez la requête en appelant la méthode `execute()`.

#### Exemple

```
$findCommand = $fm->newFindAnyCommand('Form View');  
$result = $findCommand->execute;
```

## Utilisation de la commande Rechercher

Utilisez la méthode `newFindCommand()`, en spécifiant un modèle spécifique, pour créer un objet `FileMaker_Command_Find`. Puis, exécutez la requête en appelant la méthode `execute()`.

**Remarque** Vérifiez que le nom du modèle est unique. Si votre base de données possède deux modèles du même nom, l'API FileMaker pour PHP ne pourra pas les différencier. De plus, l'API n'est pas sensible à la casse. Par exemple, si votre base de données possède un modèle appelé `SitesWeb` et un autre appelé `Sitesweb`, l'API ne pourra pas les différencier.

Utilisez la méthode `addFindCriterion()` pour ajouter des critères à la requête. Utilisez la méthode `clearFindCriteria()` pour effacer tous les critères de recherche qui ont été définis.

### Exemples

Pour rechercher un enregistrement par son nom de rubrique :

```
$findCommand = $fm->newFindCommand('Form View');
$findCommand->addFindCriterion('Questionnaire ID',
    $active_questionnaire_id);
$result = $findCommand->execute();
```

Pour indiquer un ordre de tri :

```
$findCommand = $fm->newFindCommand('Customer List');
$findCommand->addSortRule('Title', 1, FILEMAKER_SORT_ASCEND);
$result = $findCommand->execute();
```

## Utilisation d'une commande Recherche composée

La commande Recherche composée vous permet de combiner plusieurs objets de requête en une seule commande. Il existe plusieurs façons de créer une commande Recherche composée :

- Créez un objet `FileMaker_Command_CompoundFind` en appelant la méthode `newCompoundFindCommand()`.
- Créez un ou plusieurs objets `FileMaker_Command_FindRequest` en appelant la méthode `newFindRequest()`.
- Utilisez la méthode `add()` pour ajouter les objets Requête à l'objet de commande Recherche composée.
- Exécutez la commande Recherche composée en appelant la méthode `execute()`.

## Exemple

### Commande Recherche composée :

```
// Créer l'objet de commande Recherche composée
$compoundFind = $fm->newCompoundFindCommand('Form View');

// Créer la première requête
$findreq1 = $fm->newFindRequest('Form View');

// Créer la seconde requête
$findreq2 = $fm->newFindRequest('Form View');

// Créer la troisième requête
$findreq3 = $fm->newFindRequest('Form View');

// Spécifier le critère de recherche de la première requête
$findreq1->addFindCriterion('Quantity in Stock', '<100');

// Spécifier le critère de recherche de la seconde requête
$findreq2->addFindCriterion('Quantity in Stock', '0');

// Spécifier le critère de recherche de la troisième requête
$findreq3->addFindCriterion('Cover Photo Credit', 'The London Morning
News');

// Ajouter des requêtes à une commande Recherche composée
$compoundFind->add(1,$findreq1);
$compoundFind->add(2,$findreq2);
$compoundFind->add(3,$findreq3);

// Définir un ordre de tri
$compoundFind->addSortRule('Title', 1, FILEMAKER_SORT_DESCEND);

// Exécuter une commande Recherche composée
$result = $compoundFind->execute();

// Obtenir des enregistrements à partir de l'ensemble d'enregistrements
trouvés
$records = $result->getRecords();

// Indiquer le nombre d'enregistrements trouvés
echo 'Found '. count($records) . " results.<br><br>";
```

## Traitement des enregistrements d'un ensemble de résultats

- Extrayez un tableau contenant chaque enregistrement de l'ensemble de résultats en appelant la méthode `getRecords()`. Chaque membre du tableau est un objet `FileMaker_Record`, ou une instance du nom de classe défini dans l'API pour instancier les enregistrements. Le tableau peut être vide si l'ensemble de résultats ne contient aucun enregistrement.
- Obtenez la liste des noms de toutes les rubriques de l'ensemble de résultats en appelant la méthode `getFields()`. Cette méthode renvoie uniquement les noms des rubriques. Si vous avez besoin d'informations supplémentaires concernant les rubriques, utilisez l'objet de modèle lié.
- Obtenez le nombre d'enregistrements du jeu complet d'enregistrements trouvés en appelant la méthode `getFoundSetCount()`.
- Obtenez le nombre d'enregistrements du jeu d'enregistrements trouvés filtré en appelant la méthode `getFetchCount()`. Si aucun paramètre de plage n'a été spécifié pour la commande Rechercher, cette valeur est alors égale au résultat de la méthode `getFoundSetCount()`. Elle est toujours égale à la valeur de `count($response->getRecords())`.
- Pour un enregistrement spécifique, utilisez la méthode `getField()` pour renvoyer le contenu d'une rubrique sous la forme d'une chaîne.
- Pour un enregistrement spécifique, utilisez la méthode `getFieldAsTimestamp()` pour renvoyer le contenu d'une rubrique sous la forme d'un horodatage Unix (la représentation interne d'une date en PHP).
  - Si la rubrique est une rubrique Date, l'horodatage concerne la date de la rubrique à minuit.
  - Si la rubrique est une rubrique Heure, l'horodatage concerne l'heure au 1er janvier 1970.
  - Si la rubrique est une rubrique Horodatage, la valeur d'horodatage de FileMaker est directement mise en correspondance avec l'horodatage Unix.
  - Si la rubrique spécifiée n'est ni une rubrique Date ni une rubrique Heure, ou si l'horodatage généré se situe en dehors de la plage, la méthode `getFieldAsTimestamp()` renvoie un objet `FileMaker_Error`.
- Pour un enregistrement spécifique, utilisez la méthode `getContainerData()` pour renvoyer un objet de rubrique Conteneur sous la forme de données binaires :

```
<IMG src="img.php?-url=<?php echo urlencode($record->getField('Cover
Image')); ?>">
echo $fm->getContainerData($_GET['-url']);
```

- Pour un enregistrement spécifique, utilisez la méthode `getContainerDataURL()` pour renvoyer une URL entièrement qualifiée pour l'objet de la rubrique Conteneur :

```
// Pour les images, utilisez la balise HTML img
echo '';
// Pour les vidéos et les fichiers PDF, utilisez la balise HTML embed
//echo '<embed src="'. $fm->
getContainerDataURL($record->getField('container')) .'">';
```

## Limitation du nombre de lignes des tables externes renvoyées par les requêtes

Dans une solution comportant de nombreux enregistrements liés, l'interrogation et le tri des enregistrements externes peuvent demander beaucoup de temps. Pour limiter le nombre d'enregistrements affichés dans un ensemble lié, utilisez la méthode

`setRelatedSetsFilters()` avec les requêtes. La méthode `setRelatedSetsFilters()` prend deux arguments :

- Une valeur de filtre des ensembles liés : `layout` ou `none`
  - Si vous spécifiez la valeur `none`, le moteur de publication Web renvoie toutes les lignes de la table externe, ainsi que les enregistrements externes qui ne sont pas prétriés.
  - Si vous spécifiez la valeur `layout`, les paramètres indiqués dans la boîte de dialogue Table externe de FileMaker Pro sont respectés. Les enregistrements sont triés sur la base du tri défini dans la boîte de dialogue Table externe, le jeu d'enregistrements filtré commençant par la ligne initiale indiquée.
- le nombre minimum d'enregistrements de table externe renvoyés : un nombre entier ou `tout`
  - Cette valeur est utilisée uniquement lorsque le paramètre **Autoriser le défilement vertical** est activé dans la boîte de dialogue Table externe. Lorsque vous spécifiez un nombre entier, ce nombre de lignes est renvoyé après la ligne initiale. Lorsque vous spécifiez `all`, le moteur de publication Web renvoie tous les enregistrements liés.
  - Lorsque le paramètre **Autoriser le défilement vertical** est désactivé, le paramètre **Nombre de rangées** de la boîte de dialogue Table externe détermine le nombre maximal d'enregistrements liés qui sont renvoyés.

**Remarque** Le paramètre **Filtrer les enregistrements de la table externe** dans la boîte de dialogue Table externe n'est pas compatible avec les requêtes PHP. Tout calcul spécifié pour le paramètre **Filtrer les enregistrements de la table externe** est ignoré.

## Prévalidation des commandes, des enregistrements et des rubriques

La classe FileMaker vous permet de *prévalider* les données d'une rubrique dans une solution PHP sur le serveur Web avant de les valider dans la base de données.

La prévalidation est désactivée par défaut. Utilisez l'interface de ligne de commande (CLI) pour activer la prévalidation. Consultez l'Aide de l'interface de ligne de commande (CLI).

Lorsque vous envisagez d'utiliser la prévalidation, tenez compte du nombre de valeurs saisies par l'utilisateur Web. Si l'utilisateur met à jour un petit nombre de rubriques, vous pouvez améliorer la performance en n'utilisant pas la prévalidation. Mais si l'utilisateur saisit des données dans de nombreuses rubriques, la prévalidation peut lui éviter la frustration liée au rejet d'un enregistrement par la base de données en raison d'erreurs de validation.

Avec la classe FileMaker, le moteur PHP prévalide les contraintes de rubrique suivantes :

- non vide  
Les données valides sont les chaînes de caractères non vides. Les données doivent contenir au moins un caractère.
- numérique uniquement  
Les données valides contiennent des caractères numériques uniquement.
- nombre maximal de caractères  
Les données valides contiennent au plus le nombre maximal de caractères spécifié.



- année à quatre chiffres

Les données valides sont les chaînes de caractères représentant une date avec une année à quatre chiffres au format J/M/AAAA, où A est un nombre situé entre 1 et 12 inclus, J est un nombre situé entre 1 et 31 inclus et AAAA est un nombre à quatre chiffres situé entre 0001 et 4000 inclus. Par exemple, 30/1/3030 est une valeur d'année à quatre chiffres valide. Alors que 31/4/2019 est une valeur d'année à quatre chiffres non valide car le mois d'avril ne compte pas 31 jours. La validation de la date tient compte de la barre oblique (/), la barre oblique inversée (\) et du tiret (-) comme délimiteurs. Toutefois, la chaîne ne peut pas contenir différents délimiteurs. Par exemple, 30\1-2019 n'est pas valide.

- heure du jour

Les données valides sont les chaînes de caractères représentant une valeur horaire sur 12 heures dans l'un des formats suivants :

- h
- h:m
- h:m:s
- h:m:s AM/PM
- h:m AM/PM

où h est un nombre compris entre 1 et 12 inclus et m et s sont des nombres compris entre 1 et 60 inclus.

La prévalidation du moteur PHP prend en charge la vérification implicite des données de la rubrique en fonction du type de rubrique :

- date

Une rubrique définie comme une rubrique Date est validée en fonction des règles de validation de l'« année à quatre chiffres », exception faite que la valeur année peut contenir de 0 à 4 chiffres (la valeur année peut être vide). Par exemple, 30/1 est une date valide même si aucune année n'est spécifiée.

- heure

Une rubrique définie comme une rubrique Heure est validée en fonction des règles de validation de l'« heure du jour », exception faite que le composant heure (H) peut être un nombre situé entre 1 et 24 inclus pour prendre en charge les valeurs horaires sur 24 heures.

- horodatage

Une rubrique définie comme une rubrique Horodatage est validée en fonction des règles de validation de l'« heure » du composant heure et en fonction des règles de validation de la « date » du composant date.

La classe FileMaker ne peut pas prévalider toutes les options de validation de rubrique disponibles dans FileMaker Pro. Les options de validation suivantes ne peuvent pas être prévalidées car elles dépendent de l'état de toutes les données de la base de données au moment de la validation de ces données :

- valeur unique
- valeur existante
- entre
- élément de la liste de valeurs
- contrôle par le calcul

## Prévalidation des enregistrements dans une commande

Pour un objet de commande, utilisez la méthode `validate()` pour valider une rubrique ou la commande entière en fonction des règles de prévalidation mises en œuvre par le moteur PHP. Si vous transmettez l'argument facultatif du nom de la rubrique, seule cette rubrique sera prévalidée.

Lorsque la prévalidation est réussie, la méthode `validate()` renvoie `TRUE`. Lorsque la prévalidation échoue, la méthode `validate()` renvoie un objet `FileMaker_Error_Validation` contenant des détails sur ce qui n'a pas pu être validé.

## Prévalidation des enregistrements

Pour un objet d'enregistrement, utilisez la méthode `validate()` pour valider une rubrique ou toutes les rubriques de l'enregistrement en fonction des règles de prévalidation mises en œuvre par le moteur PHP. Si vous transmettez l'argument facultatif du nom de la rubrique, seule cette rubrique sera prévalidée.

Lorsque la prévalidation est réussie, la méthode `validate()` renvoie `TRUE`. Lorsque la prévalidation échoue, la méthode `validate()` renvoie un objet `FileMaker_Error_Validation` contenant des détails sur ce qui n'a pas pu être validé.

## Prévalidation des rubriques

Pour un objet rubrique, utilisez la méthode `validate()` pour déterminer si la valeur donnée d'une rubrique est valide.

Lorsque la prévalidation est réussie, la méthode `validate()` renvoie `TRUE`. Lorsque la prévalidation échoue, la méthode `validate()` renvoie un objet `FileMaker_Error_Validation` contenant des détails sur ce qui n'a pas pu être validé.

## Traitement des erreurs de validation

Lorsque la prévalidation échoue, l'objet `FileMaker_Error_Validation` renvoyé contient un tableau comportant trois éléments pour chaque échec de validation :

1. L'objet de rubrique pour lequel la prévalidation a échoué
2. Une constante de validation indiquant la règle de validation qui a échoué :
  - 1 - `FILEMAKER_RULE_NOTEMPTY`
  - 2 - `FILEMAKER_RULE_NUMERICONLY`
  - 3 - `FILEMAKER_RULE_MAXCHARACTERS`
  - 4 - `FILEMAKER_RULE_FOURDIGITYEAR`
  - 5 - `FILEMAKER_RULE_TIMEOFDAY`
  - 6 - `FILEMAKER_RULE_TIMESTAMP_FIELD`
  - 7 - `FILEMAKER_RULE_DATE_FIELD`
  - 8 - `FILEMAKER_RULE_TIME_FIELD`
3. La valeur réelle entrée dans la rubrique pour laquelle la prévalidation a échoué

Vous pouvez également utiliser les méthodes suivantes avec un objet `FileMaker_Error_Validation` :

- Utilisez la méthode `isValidatationError()` pour savoir si une erreur est une erreur de validation.
- Utilisez la méthode `numErrors()` pour obtenir le nombre de règles de validation ayant échoué.

### Exemple

```
//Créer une requête Ajouter
$addrequest = $fm->newAddCommand('test', array('join' => 'added', 'maxchars'
=> 'abcx', 'field' => 'something' , 'numericonly' => 'abc'));

//Valider toutes les rubriques
$result = $addrequest->validate();

//Si la méthode validate() a renvoyé des erreurs, afficher le nom de la
rubrique, le numéro d'erreur et la valeur ayant échoué.
if(FileMaker::isError($result)){
    echo 'Validation failed:'. "\n";
    $validationErrors= $result->getErrors();
    foreach ($validationErrors as $error) {
        $field = $error[0];
        echo 'Field Name: " . $field->getName(). "\n";
        echo 'Error Code: " . $error[1] . "\n";
        echo 'Value: " . $error[2] . "\n";
    }
}
```

### Résultat

```
Validation failed:
Field Name: numericonly
Error Code: 2
Value: abc
Field Name: maxchars
Error Code: 3
Value: abcx
```

## Gestion des erreurs

La classe FileMaker définit l'objet FileMaker\_Error pour vous permettre de gérer les erreurs qui se produisent dans une solution PHP.

Une erreur peut se produire lorsqu'une commande est exécutée. Si une erreur se produit, la commande renvoie un objet FileMaker\_Error. Il est conseillé de vérifier l'erreur renvoyée lorsqu'une commande est exécutée.

Utilisez les méthodes suivantes pour en savoir plus sur l'erreur indiquée dans l'objet FileMaker\_Error.

- Vérifiez si une variable est un objet FileMaker Error en appelant la méthode `isError()`.
- Obtenez le nombre d'erreurs qui se sont produites en appelant la méthode `numErrors()`.
- Extrayez un ensemble de tableaux décrivant les erreurs qui se sont produites en appelant la méthode `getErrors()`.
- Affichez un message d'erreur en appelant la méthode `getMessage()`.

### Exemple

```
$result = $findCommand->execute();
if (FileMaker::isError($result)) {
    echo "<p>Error: " . $result->getMessage() . "</p>";
    exit;
}
```

Pour plus d'informations sur les codes d'erreur renvoyés avec l'objet FileMaker Error, consultez l'annexe A, « Codes d'erreur de la Publication Web personnalisée ».

# Chapitre 9

## Stockage, test et suivi d'un site

Ce chapitre propose des instructions pour stocker et tester un site en Publication Web personnalisée avant son déploiement dans un environnement de production. Vous y trouverez également des instructions relatives à l'utilisation des fichiers journaux permettant d'assurer le suivi du site en phase de test ou après le déploiement.

### Stockage d'un site de Publication Web personnalisée

Avant de pouvoir tester correctement votre site, vous devez copier ou déplacer les fichiers nécessaires à leur emplacement correct sur le(s) serveur(s) de stockage.

1. Suivez toutes les étapes exposées dans chapitre 2, « Préparation des bases de données pour la Publication Web personnalisée ».
2. Assurez-vous que la Publication Web personnalisée a été activée et correctement paramétrée dans l'interface de ligne de commande (CLI). Consultez l'[Aide FileMaker Server](#).
3. Vérifiez le bon fonctionnement du serveur Web et du moteur de publication Web.
4. Copiez ou déplacez les fichiers de votre site vers le composant serveur Web de votre déploiement FileMaker Server.

Copiez ou déplacez les fichiers de votre site vers le répertoire suivant :

- IIS (Windows) via HTTP ou HTTPS :  
`[disque]:\Program Files\FileMaker\FileMaker Server\HTTPServer\Conf`  
où [disque] correspond au disque sur lequel réside votre ordinateur maître FileMaker Server.
- Apache (macOS) via HTTP :  
`/Bibliothèque/FileMaker Server/HTTPServer/htdocs`
- Apache (macOS) via HTTPS :  
`/Bibliothèque/FileMaker Server/HTTPServer/htdocs/httpsRoot`

**Remarque** Lorsque vous importez un certificat SSL personnalisé, les connexions de clients au serveur de base de données utilisent SSL et les connexions HTTP sont acheminées vers HTTPS. Avec un certificat SSL personnalisé, utilisez le répertoire HTTPS pour l'hébergement des fichiers PHP du site.

5. Si ce n'est pas déjà fait, copiez ou déplacez tout objet de rubrique Conteneur référencé vers l'emplacement correct sur l'ordinateur maître.
  - Si le fichier de base de données est correctement hébergé et accessible sur le composant Database Server du déploiement de FileMaker Server, et si les types de données de conteneur stockent les fichiers proprement dits dans la base de données FileMaker Pro, toute action sur le contenu de la rubrique Conteneur est alors inutile.
  - Si une rubrique de base de données Conteneur stocke une référence à un fichier et non le fichier lui-même, l'objet Conteneur référencé doit alors être stocké dans le dossier Web de FileMaker Pro lorsque l'enregistrement est créé ou modifié. Pour stocker votre site, vous devez copier ou déplacer les objets Conteneur référencés dans un dossier possédant le même emplacement relatif dans le dossier racine du logiciel de serveur Web.
  - Lorsque vous utilisez FileMaker pour télécharger une base de données avec une rubrique Conteneur stockant des objets en externe, les données de la rubrique Conteneur stockées en externe sont téléchargées vers FileMaker Server dans le cadre du processus. Pour en savoir plus sur le transfert de fichiers de base de données vers FileMaker Server, consultez l'[Aide FileMaker Pro](#).
  - Lorsque vous téléchargez manuellement une base de données utilisant une rubrique Conteneur avec des objets stockés en externe, vous devez copier ou déplacer les objets référencés dans un sous-dossier du dossier RC\_Data\_FMS, comme décrit dans la section « Types de données de conteneur avec données stockées en externe », page 17.
6. Copiez les composants supplémentaires de votre application Web sur l'ordinateur maître. Pour la Publication Web personnalisée avec XML, votre application Web traite les données XML avant de les envoyer vers une autre application ou au client.

## Test d'un site de Publication Web personnalisée

Avant d'informer les utilisateurs de la disponibilité de votre site de Publication Web personnalisée, vérifiez s'il se présente et fonctionne comme prévu.

- Testez les fonctions telles que la recherche, l'ajout, la suppression et le tri d'enregistrements avec différents jeux de comptes et de privilèges.
- Vérifiez si les jeux de privilèges fonctionnent comme prévu en vous connectant à différents comptes. Faites en sorte que des utilisateurs non autorisés ne puissent pas accéder à vos données ni les modifier.
- Contrôlez tous les scripts pour vérifier que leur effet est conforme aux attentes. Consultez la section « Scripts FileMaker et Publication Web personnalisée », page 18 pour obtenir des informations sur la conception de scripts adaptés au Web.
- Testez votre site avec différents systèmes d'exploitation et navigateurs Web.
- Lors de la création de solutions utilisant l'API FileMaker pour PHP, nous recommandons la conception de solutions prenant en charge les cookies. L'API FileMaker pour PHP offre des délais de réponse plus courts lorsque les cookies sont activés. Les cookies ne sont pas nécessaires à l'utilisation des fonctions de Publication Web personnalisée, mais permettant la mise en cache des informations de session par le moteur de publication Web.

**Remarque** Vous pouvez afficher et tester votre site sur l'ordinateur maître sans pour autant utiliser de connexion réseau, par exemple en saisissant `http://127.0.0.1/` dans l'URL.

- Pour les solutions PHP, utilisez `http://127.0.0.1/<chemin_site>` où `<chemin_site>` est le chemin relatif vers la page d'accueil de votre site.
- Pour plus d'informations sur la syntaxe d'URL dans les solutions XML, consultez la section « À propos de la syntaxe d'URL pour les données XML et les objets de type Conteneur », page 27.

## Feuilles de style permettant de tester les sorties XML

### Exemples

Voici deux exemples de feuilles de style XSLT utiles pour tester le résultat XML.

L'exemple de feuille de style suivant renvoie les données XML requises sans aucune transformation. Cette feuille de style est utile pour afficher les données XML réelles utilisées par le moteur de publication Web.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
  <xsl:output method="xml"/>
  <xsl:template match="/">
    <xsl:copy-of select="."/>
  </xsl:template>
</xsl:stylesheet>
```

Pour le débogage d'une feuille de style, vous pouvez utiliser l'exemple suivant d'une balise HTML `<textarea>` pour afficher le document source XML accessible via la feuille de style, dans une zone de texte défilante.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
  <xsl:output method="html"/>
  <html>
    <body>
      <xsl:template match="/fmrs:fmresultset">
        <textarea rows="20" cols="100">
          <xsl:copy-of select="."/>
        </textarea><br/>
      </xsl:template>
    </body>
  </html>
</xsl:stylesheet>
```

## Suivi de votre site

Vous pouvez utiliser les types de fichiers de consignation suivants pour surveiller votre site de Publication Web personnalisée et réunir des informations au sujet des utilisateurs Web qui visitent votre site :

- Journaux d'erreurs et d'accès au serveur Web
- Journal du moteur de publication Web
- Journal d'erreurs du module de serveur Web
- Journaux Tomcat

### Utilisation des journaux d'erreurs et d'accès au serveur Web

IIS (Windows) : Le serveur Web Microsoft IIS génère un fichier journal des accès, mais affiche les erreurs dans l'Observateur d'événements de Windows au lieu de les consigner dans un fichier journal. Le fichier journal des accès, qui répond par défaut au format étendu de fichier journal du W3C, consigne toutes les requêtes HTTP reçues par le serveur Web. Vous pouvez aussi utiliser le format standard de fichier journal du W3C pour le journal des accès. Consultez la documentation du serveur Web Microsoft IIS.

Apache (macOS) : Le serveur Web Apache génère un fichier journal des accès et un fichier journal des erreurs. Le fichier journal des accès Apache, qui répond par défaut au format standard de fichier journal du W3C, consigne toutes les requêtes HTTP reçues par le serveur Web. Le journal des erreurs Apache consigne les problèmes survenus dans le cadre du traitement des requêtes HTTP. Consultez la documentation du serveur Web Apache.

**Remarque** Pour obtenir des informations sur les formats standard et étendu de fichier journal du W3C, consultez le site Web du World Wide Web Consortium, à l'adresse [www.w3.org](http://www.w3.org).

### Utilisation du journal du moteur de publication Web

Le moteur de publication Web génère un fichier journal nommé `wpe.log` qui contient un enregistrement des erreurs de moteur de publication Web qui se sont produites, notamment les erreurs d'application, d'utilisation et système.

Le fichier `wpe.log` est généré si le moteur de publication Web est en cours d'exécution. (Consultez l'Admin Console pour vérifier si le moteur de publication Web est en cours d'exécution.)

Le fichier `wpe.log` se trouve sur le composant Moteur de publication Web du déploiement FileMaker Server :

- **Windows :**  
`[disque]:\Program Files\FileMaker\FileMaker Server\Logs\wpe.log`  
où `[disque]` correspond au lecteur principal servant au démarrage de votre système.
- **macOS :** `/Bibliothèque/FileMaker Server/Logs/wpe.log`

### Format du journal du moteur de publication Web

Le fichier `wpe.log` utilise le format suivant pour chaque entrée :

```
[TIMESTAMP_GMT] [WPC_HOSTNAME] [CLIENT_IP:PORT] [ACCOUNT_NAME] [MODULE_TYPE]  
[SEVERITY] [FM_ERRORCODE] [RETURN_BYTES] [MESSAGE]
```

où :

- `[TIMESTAMP_GMT]` correspond aux date et heure de la saisie, à l'heure de Greenwich (GMT).
- `[WPC_HOSTNAME]` est le nom de l'ordinateur maître.
- `[CLIENT_IP:PORT]` est l'adresse IP et le port client d'origine de la requête XML.



- [ACCOUNT\_NAME] est le nom de compte utilisé pour se connecter à la base de données FileMaker Pro hébergée.
- [MODULE\_TYPE] est soit : XML, pour la Publication Web personnalisée, ou PHP, pour la Publication Web personnalisée avec requête PHP.
- [SEVERITY] est soit INFO, désignant un message d'information ou ERREUR, désignant un message d'erreur.
- [FM\_ERROR\_CODE] est le numéro d'erreur renvoyé pour un message d'erreur. Le numéro d'erreur peut être un code erreur des bases de données FileMaker Pro (consultez la section « Numéros des codes d'erreur des bases de données FileMaker Pro », page 100).  
En outre, le numéro d'erreur peut être un numéro HTTP, précédé de la chaîne « HTTP: »
- [RETURN\_BYTES] est le nombre d'octets renvoyés par la requête.
- [MESSAGE] fournit des informations complémentaires sur l'entrée du journal.

## Message de journal de moteur de publication Web

### Exemples

Les exemples suivants affichent les types de messages pouvant être inclus dans le fichier wpe.log.

#### Lorsque le moteur de publication Web démarre et s'arrête :

```
2019-06-02 15:15:31 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine started.
2019-06-02 15:46:52 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine stopped.
```

#### Requêtes XML ayant abouti ou échoué :

```
2019-06-02 15:21:08 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML INFO
0 3964 "/fmi/xml/fmresultset.xml?-db=Contacts&-lay=Contact_Details&-
findall"
2019-06-02 15:26:31 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML
ERROR 5 596 "/fmi/xml/fmresultset.xml?-db=Contacts&-
layout=Contact_Details&-findall"
```

#### Erreurs de création de script :

```
2019-06-02 17:33:12 -0700 WPC_SERVER 192.168.100.101:0 jdoe - ERROR
4 - Web Scripting Error: 4, File: "10b_MeetingsUpload", Script: "OnOpen",
Script Step: "Show Custom Dialog"
```

#### Modification des paramètres de Publication Web personnalisée :

```
2019-06-09 10:59:49 -0700 WPC_SERVER 192.168.100.101:0 jdoe - INFO
- - XML Web Publishing Engine is enabled.
```

#### Erreurs système :

```
2019-06-02 15:30:42 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML
ERROR - - Communication failed
```

## Utilisation du journal des erreurs du module de serveur Web

Si le serveur Web n'est pas en mesure de se connecter au moteur de publication Web, le module de serveur Web génère un fichier journal consignnant toutes les erreurs liées à son utilisation. Ce fichier se nomme `web_server_module_log.txt` et se situe dans le sous-dossier Logs du dossier FileMaker Server, sur l'hôte du serveur Web.

## Utilisation des journaux Tomcat

Lorsque FileMaker Server a un problème causé par une erreur de serveur Web interne, il peut être utile d'afficher les journaux Tomcat. Les journaux Tomcat sont affectés sur le composant de serveur Web du déploiement de FileMaker Server :

- Windows :
  - `[drive]:\Program Files\FileMaker\FileMaker Server\Admin\admin-master-tomcat\logs\`  
où [disque] correspond au lecteur principal servant au démarrage de votre système.
  - `[disque]:\Program Files\FileMaker\FileMaker Server\Web Publishing\publishing-engine\jwpc-tomcat\logs`  
où [disque] correspond au lecteur principal servant au démarrage de votre système.
- macOS :
  - `/Bibliothèque/FileMaker Server/Admin/admin-master-tomcat/logs/`
  - `/Bibliothèque/FileMaker Server/Web Publishing/publishing-engine/jwpc-tomcat/logs`

# Annexe A

## Codes d'erreur de la Publication Web personnalisée

Le moteur de publication Web génère des codes d'erreur de base de données et les erreurs de chaîne de requête qui peuvent se produire pendant une requête de données XML.

Pour obtenir la liste des codes d'erreur mis à jour, reportez-vous à la [Base de connaissances](#).

### Numéros de code d'erreur au format XML

Le moteur de publication Web génère un code d'erreur pour les bases de données publiées au format XML lorsque des données sont requises. Ce type de valeur de code d'erreur s'insère au début du document XML dans l'élément `<error code>` pour la grammaire `fmresultset` ou dans l'élément `<ERRORCODE>` pour les grammaires `FMPXMLRESULT` ou `FMPXMLLAYOUT`. Un code d'erreur égal à 0 indique qu'aucune erreur ne s'est produite.

#### Exemples

**Code d'erreur de la base de données dans la grammaire `fmresultset` :**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fmresultset PUBLIC "-//FMI//DTD fmresultset//EN"
"http://192.168.123.101/fmi/xml/fmresultset.dtd">
<fmresultset xmlns="http://www.filemaker.com/xml/fmresultset"
version="1.0">
  <error code="0"></error>
```

**Code d'erreur de la base de données dans la grammaire `FMPXMLRESULT` :**

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE FMPXMLRESULT PUBLIC "-//FMI//DTD FMPXMLRESULT//EN"
"http://192.168.123.101/fmi/xml/FMPXMLRESULT.dtd">
<fmpxmlresult xmlns="http://www.filemaker.com/fmpxmlresult">
  <ERRORCODE>0</ERRORCODE>
```

En tant que développeur de la solution de publication Web personnalisée, il vous appartient de vérifier la valeur de l'élément `<error code>` ou `<ERRORCODE>` et d'agir en conséquence. Le moteur de publication Web ne gère pas les erreurs de base de données.

## Numéros des codes d'erreur des bases de données FileMaker Pro

Pour obtenir les codes d'erreur de FileMaker Pro, consultez l'[Aide FileMaker Pro](#).

FileMaker Server renvoie le code d'erreur 959 pour indiquer qu'une technologie a été désactivée.

Par exemple, si l'administrateur serveur désactive la Publication Web personnalisée en utilisant l'interface de ligne de commande (CLI), les requêtes XML renvoient le code d'erreur 959.

### Exemple

Code d'erreur 959 dans la grammaire FMPXMLLAYOUT :

```
<FMPXMLLAYOUT>
  <ERRORCODE>959</ERRORCODE>
  <LAYOUT DATABASE="" NAME="" />
  <VALUELISTS />
</FMPXMLLAYOUT>
```

# Index

## A

- action de script Reconnexion 14
- actions de script
  - reconnexion 14
- actions de scripts
  - modifier le mot de passe 14
- activation de la Publication Web personnalisée dans une base de données 13
- add(), méthode 85
- addSortRule(), méthode 83
- Admin Console 15
- adresse IP dynamique 66
- adresse IP statique 66
- ajouter, commande 75
- année à quatre chiffres, rubrique 89
- API FileMaker pour PHP 11
  - installation manuelle 66
  - référence 71
- authentification de base, Web 14
- authentification des utilisateurs Web 14
- auto-enter, attribut 32
- autoriser le défilement vertical, paramètre 58, 88
- avantages de PHP 12

## B

- bases de données, protection après publication 15
- bibliothèque d'URL du client 65

## C

- caractères ASCII, dans les documents XML 39
- chaînes de requête
  - ajout d'enregistrements à des tables externes 46
  - commandes et paramètres 39, 43
  - consignes 44
  - modification d'enregistrements dans des tables externes 47
  - nom de rubrique entièrement qualifié, syntaxe 46
  - rubriques de type Global, syntaxe 48
- chaînes de requêtes
  - demande de données XML 39, 43
- classe FileMaker 72
- clearSortRules(), méthode 83
- CLI, interface de ligne de commande 15
- codage
  - données PHP 70
  - données XML 29, 39
  - URL 29
- codage de texte
  - données XML générées 29
  - URL 29
- codage de texte URL 29
- codes d'erreurs des bases de données 30, 99
- commandes. *Voir* chaînes requête

- commit(), méthode 74
- comparaison des grammaires XML 29
- compte Invité
  - activation 14
  - avec la Publication Web personnalisée 14
  - désactivation 14
- comptes et privilèges
  - activation de la Publication Web personnalisée 13
  - compte Invité 14
  - scripts 19
- configuration de la Publication Web personnalisée 22
- configuration du serveur 65
- connexion à une base de données FileMaker Pro à l'aide du langage PHP 73
- contrôle par le calcul 89
- createRecord(), méthode 74
- création d'un enregistrement
  - à l'aide du langage PHP 74
  - à l'aide du langage XML 51
- cryptage SSL (Secure Sockets Layer) 15
- cURL 65

## D

- <datasource> élément 31
- date, rubrique 89
- db, paramètre de requête 52
- dbnames, commande de requête 49
- définitions de type de document (DTD) 30, 34
- delete, commande de requête 49
- delete(), méthode 76, 81
- dépannage
  - accès aux documents XML 42
  - sites de Publication Web personnalisée 94
- des déclencheurs de script 20
- documentation en ligne 8
- documentation FileMaker Server 8
- documentation, produit FileMaker 8
- dossier Web, copie des objets de rubrique Conteneur 16
- dup, commande de requête 49
- duplication d'un enregistrement 75
- dupliquer, commande 75

## E

- édition d'un enregistrement 75
- edition, commande 75
- élément de la liste de valeurs, validation 89
- et, éléments 99
- éléments
  - code d'erreur des bases de données 30
  - dans la grammaire FMPXMLLAYOUT 36
  - dans la grammaire FMPXMLRESULT 34
  - dans la grammaire fmresultset 31

enregistrements  
 créer dans l'PHP 74  
 créer dans l'XML 51  
 dupliquer dans l'PHP 75  
 dupliquer dans l'XML 49  
 ignorer dans l'XML 62  
 modifier dans l'PHP 75  
 modifier dans l'XML 49  
 rechercher dans l'PHP 83  
 rechercher dans l'XML 50  
 supprimer dans l'PHP 76  
 supprimer dans l'XML 49

ensemble de résultats 87

entre, validation 89

erreurs  
 décrites 99  
 éléments des codes d'erreurs des bases de données 30  
 fichiers journaux du serveur Web 96  
 gestion 92  
 numéros des codes d'erreur des bases de données 99

espaces de nom pour XML 30

esperluette, caractères dans PHP 73

exécution de requêtes 83

exemples  
 grammaire FMPXMLLAYOUT générée 38  
 grammaire FMPXMLRESULT générée 35  
 grammaire fmresultset générée 33

exportation de données XML 24

Extensible Markup Language (XML). *Voir* XML

**F**

feuilles de style, test 94

fichiers journaux 94  
 accès au serveur Web 96  
 description 96  
 Tomcat 98  
 web\_server\_module\_log.txt 98

fichiers journaux d'accès au serveur Web, description 96

fichiers PDF 8

field name, paramètre de requête (non-conteneur) 53

-field, paramètre de requête (Conteneur) 53

<field-definition>, élément 32

-fieldname.op, paramètre de requête 54

FileMaker Pro, différence avec le moteur de Publication Web 24

FileMaker WebDirect 9

filtrage des enregistrements des rubriques de table externe 58

filtrer les enregistrements de la table externe, paramètre 59, 88

-find, commande de requête 50

-findall, commande de requête 50

-findany, commande de requête 50

-findquery, commande de requête 50

fmphp, mot-clé pour l'activation de la publication PHP 13

FMPXMLLAYOUT, grammaire 24, 29, 36–38

FMPXMLRESULT, grammaire 24, 29, 34–35

fmresultset, grammaire 24, 29, 30–33

fmsadmin, groupe 17

fmxml, mot-clé pour l'activation de la publication XML 13, 26

format UTF-8 (Unicode Transformation 8 Bit) 29, 39

formulaire HTML de requêtes XML 26

four-digit-year, attribut 32

## G

gestion des erreurs 92

getDatabase(), méthode 79

getErrors(), méthode 92

getFetchCount(), méthode 87

getField(), méthode 87

getFieldAsTimestamp(), méthode 87

getFields(), méthode 79, 87

getFoundSetCount(), méthode 87

getLayout(), méthode 79

getMessage(), méthode 92

getName(), méthode 79, 81

getRange(), méthode 84

getRecords(), méthode 87

getRelatedSet(), méthode 80

getRelatedSets(), méthode 80

getValueListsTwoFields(), méthode 82

getValueListTwoFields(), méthode 82

grammaires pour XML, description 29

## H

heure du jour, rubrique 89

heure, rubrique 89

horodatage Unix 87

horodatage, rubrique 87, 89

hostspect, propriété pour PHP 74

HTTPS 16  
 avec rubrique Conteneur 18  
 protocole en syntaxe d'URL 27  
 répertoire sécurisé pour macOS 67

## I

importation de données XML 24

informations de référence 71

installation de l'API FileMaker pour PHP 66

installation manuelle de l'API FileMaker pour PHP 66

interface de ligne de commande (CLI) 15

isError(), méthode 92

isValidationError(), méthode 91

## J

jeu de privilèges d'accès, publication Web personnalisée 13

journal d'application 96

## L

L'AdminConsole de FileMaker Server 15

la commande de requête -edit 49

Latin-1 encoding 70

-lay, paramètre de requête 41, 55

-lay.response, paramètre de requête 41, 55

- layoutnames, commande de requête 51
- limite des enregistrements de rubrique de table externe 59
- listes de valeurs
  - utilisation dans PHP 82
  - utilisation dans XML 36
- listFields(), méthode 79
- listLayouts(), méthode 79
- listRelatedSets(), méthode 79, 80
- listScripts(), méthode 76
- listValueLists(), méthode 79, 82
- lop, paramètre de requête 56

## M

- macOS Server Admin 65
- max, paramètre de requête 56
- max-characters, attribut 32
- max-repeat, attribut 32
- <metadata> élément 32
- méthode getContainerData() 87
- méthode getContainerDataURL() 87
- méthodes
  - add() 85
  - addSortRule() 83
  - clearSortRules() 83
  - commit() 74
  - createRecord() 74
  - delete() 76, 81
  - getContainerData() 87
  - getContainerDataURL() 87
  - getDatabase() 79
  - getErrors() 92
  - getFetchCount() 87
  - getField() 87
  - getFieldAsTimestamp() 87
  - getFields() 79, 87
  - getFoundSetCount() 87
  - getLayout() 79
  - getMessage() 92
  - getName() 79, 81
  - getRange() 84
  - getRecords() 87
  - getRelatedSet() 80
  - getRelatedSets() 80
  - getValueListsTwoFields() 82
  - getValueListTwoFields() 82
  - isError() 92
  - isValidationError() 91
  - listFields() 79
  - listLayouts() 79
  - listRelatedSets() 79, 80
  - listScripts() 76
  - listValueLists() 79, 82
  - newAddCommand() 75
  - newCompoundFindCommand() 85
  - newDeleteCommand() 76
  - newDuplicateCommand() 75
  - newEditCommand() 75
  - newFindAllCommand() 84

- newFindAnyCommand() 84
- newFindCommand() 85
- newFindRequest() 85
- newPerformScriptCommand() 77
- newRelatedRecord() 81
- numErrors() 91, 92
- setLogicalOperator() 83
- setPreCommandScript() 77, 84
- setPreSortScript() 77, 84
- setProperty() 73
- setRange() 84
- setRelatedSetsFilters() 88
- setResultsLayout() 79
- setScript() 77, 84
- validate() 90
- MIME (Multipurpose Internet Mail Extensions), types 16
- modèles
  - passage de l'un à l'autre pour une réponse XML 41
  - utilisation dans le PHP 79
- modèles de base de données disponibles 51
- modid, paramètre de requête 56
- modifier le mot de passe, script 14
- moteur de Publication Web
  - avantages du 21
  - description 10
  - génération de documents XML 26
  - génération de données XML 25
  - journal d'application 96
  - traitement d'une requête 10
- moteur de publication Web
  - codes d'erreurs générés 99
- moteur de Publication Web personnalisée (CWPE) 25
- mots de passe
  - aucun mot de passe de connexion 14
  - authentification de base pour les utilisateurs Web 14
  - définition pour la Publication Web personnalisée 13
  - script Modifier le mot de passe 14
- mots-clés pour l'activation de la Publication Web personnalisée 13, 26

## N

- name, attribut 32
- new, commande de requête 51
- newAddCommand(), méthode 75
- newCompoundFindCommand(), méthode 85
- newDeleteCommand(), méthode 76
- newDuplicateCommand(), méthode 75
- newEditCommand(), méthode 75
- newFindAllCommand(), méthode 84
- newFindAnyCommand(), méthode 84
- newFindCommand(), méthode 85
- newFindRequest(), méthode 85
- newPerformScriptCommand(), méthode 77
- newRelatedRecord(), méthode 81
- nom de rubrique entièrement qualifié, syntaxe 46
- Nombre de rangées, paramètre 58, 88
- nombre maximal de caractères, rubrique 88

- noms d'utilisateur
  - authentification de base pour les utilisateurs Web 14
- noms d'utilisateurs
  - définition pour la Publication Web personnalisée 13
- noms de rubrique, syntaxe entièrement qualifiée 46
- non vide, rubrique 88
- not-empty, attribut 32
- nouvelles fonctionnalités de la Publication Web personnalisée 22
- noyau de Publication Web illustré 25
- numeric-only, attribut 32
- numérique uniquement, rubrique 88
- numéros des codes d'erreurs des bases de données 99
- numErrors(), méthode 91, 92

## O

- objet d'enregistrement 74
- objet d'ensemble lié 80
- objet de base de données 73
- objets de classe FileMaker
  - base de données 73
  - définition 72
  - enregistrement 74
  - ensemble lié 80
- objets de commande FileMaker
  - ajouter 75
  - dupliquer 75
  - édition 75
  - recherche composée 85
  - Rechercher 83
  - rechercher 85
  - rechercher au moins 84
  - rechercher tout 84
  - supprimer 76
- opérateurs de comparaison des rubriques 54
- opérateurs, comparaison 54
- ordre de traitement de requête XML 42
- outil Server Admin. *Voir* macOS Server Admin

## P

- page d'accueil du site PHP 69
- paramètres des requêtes. *Voir* chaînes requête
- passage d'un modèle à un autre pour une réponse XML 41
- PHP
  - activation dans une base de données 13
  - page d'accueil du site 69
  - publication Web personnalisée, décrite 11
  - version prise en charge 66
- plug-ins 19
- plus de mémoire, erreur 19
- présentation
  - publication Web personnalisée 9
- présentation générale
  - étapes pour accéder aux données XML 26

- prévalidation
  - année à quatre chiffres 89
  - commandes 88
  - date 89
  - enregistrements 90
  - heure 89
  - heure du jour 89
  - horodatage 89
  - nombre maximal de caractères 88
  - non vide 88
  - numérique uniquement 88
  - rubriques 90
- privilège d'accès étendu, publication Web personnalisée 13
- privilèges d'accès 14
- protection des bases de données publiées 15
- publication statique, décrite 9
- publication sur le Web
  - codes d'erreurs des bases de données 99
  - configuration requise pour 22
  - connexion à Internet ou à un intranet 23
  - objets de rubrique Conteneur 16
  - protection des bases de données 15
  - utilisation de données XML 26
- Publication Web personnalisée
  - avec XML 11, 24
- publication Web personnalisée
  - accès aux solutions par les utilisateurs Web 14
  - activation dans le moteur de Publication Web 15
  - activation dans une base de données 13
  - avec PHP 11
  - compte Invité 14
  - configuration requise pour 22
  - définition 9
  - nouvelles fonctionnalités dans 22
  - plug-ins pour les solutions de Publication Web 19
  - présentation 9
  - privilège étendu pour 13
  - restriction d'accès au serveur Web à l'aide des adresses IP 15
  - scripts 20
  - utilisation d'une adresse IP statique 23
  - utilisation de scripts 18

## Q

- query, paramètre de requête 57

## R

- rangée initiale, paramètre 58
- récapitulatif des étapes pour accéder aux données XML 26
- recherche composée
  - commande 85
  - commande de requête 50
  - exemple 86
  - paramètre de requête 57
- rechercher au moins, commande 84
- rechercher tout, commande 84
- rechercher, commande 85
- rechercher, objets de commande 83



- recid, paramètre de requête 58
  - récupération
    - informations de modèle 52
    - noms de scripts disponibles 52
    - noms des modèles 51
  - <relatedset-definition>, élément 32
  - relatedsets.filter, paramètre de requête 47, 58
  - relatedsets.max, paramètre de requête 59
  - répertoire HTTP pour PHP 69
  - répertoire HTTPS pour PHP 69
  - représentation de la date 87
  - requêtes de rubriques de table externe 48, 58, 59
  - requêtes pour des données XML 26
  - result, attribut 32
  - <resultset> élément 32
  - rôle du navigateur Web dans les requêtes XML 25
  - rubriques
    - année à quatre chiffres 89
    - attributs 32
    - calcul 29, 30
    - conteneur 16, 27, 34
    - date 34, 37, 89
    - global 30
    - heure 34, 37, 89
    - heure du jour 89
    - horodatage 34, 37, 89
    - lié au PHP 80
    - lié au XML 32, 47
    - multivaluées 28, 46
    - nombre 34
    - nombre maximal de caractères 88
    - noms de rubrique entièrement qualifiés 46
    - non vide 88
    - numérique uniquement 88
    - Statistique 29, 30
    - table externe 32
    - tables externes 46
    - texte 34
  - Rubriques Conteneur
    - avec données stockées en externe 17
  - rubriques Conteneur
    - accès aux données par les utilisateurs Web 18
    - avec fichiers référencés 16
    - publication du contenu de 16
    - syntaxe d'URL pour accéder aux solutions XML 27
    - téléchargement progressif 18
  - rubriques de type Global
    - syntaxe 48
  - rubriques globales
    - dans la définition de rubrique 32
- S**
- SAT. *Voir* Admin Console de FileMaker Server
  - script, paramètre de requête 60
  - script.param, paramètre de requête 60
  - script.prefind, paramètre de requête 60
  - script.prefind.param, paramètre de requête 61
  - script.presort, paramètre de requête 61
  - scriptnames, commande de requête 52
  - scripts
    - astuces et considérations 19
    - comptes et privilèges 19
    - dans la Publication Web personnalisée 18
    - utilisation dans le PHP 76
    - utilisation dans les requêtes XML 26
  - scripts disponibles 52
  - sécurité
    - comptes et mots de passe 15
    - consignes pour la protection de vos bases de données publiées 15
    - documentation 11
    - limitation d'accès depuis des adresses IP 15
  - serveur Web
    - fichiers journaux 96
    - prise en charge du type MIME 16
    - rôle dans les requêtes XML 25
  - setLogicalOperator(), méthode 83
  - setPreCommandScript(), méthode 77, 84
  - setPreSortScript(), méthode 77, 84
  - setProperty(), méthode 73
  - setRange(), méthode 84
  - setRelatedSetsFilters(), méthode 88
  - setResultsLayout(), méthode 79
  - setScript(), méthode 77, 84
  - sites Web
    - création à l'aide du moteur de Publication Web 21
    - pages d'assistance 8
    - suiwi 96
    - test 94
  - skip, paramètre de requête 62
  - sortfield, paramètre de requête 62
  - sortorder, paramètre de requête 63
  - spécification du modèle au moment d'effectuer une requête sur les données XML 41
  - suiwi des sites Web 96
  - suppression d'enregistrements dans une table externe 47
  - suppression d'un enregistrement 76
  - supprimer, commande 76
  - syntaxe d'URL pour
    - objets Conteneur dans les solutions XML 27
    - requêtes XML 27
- T**
- tables externes
    - ajout d'enregistrements 46
    - modèle 58
    - modifications d'enregistrements 47
    - nombre d'enregistrements 58
    - rangée initiale 58
    - suppression d'enregistrements 47
    - tri d'enregistrements 59
    - utilisation dans le PHP 80
    - téléchargement progressif 18

test  
 sites Web 94  
 sorties XML 95  
 time-of-day, attribut 32  
 Tomcat, utilisation de fichiers journaux 98  
 traitement d'un ensemble de résultats 87  
 traitement d'une requête du moteur de publication Web 10  
 tri des enregistrements de rubrique de table externe 59  
 type, attribut 32

## U

UAC. *Voir* Admin Console de FileMaker Server  
 Unicode  
 caractères utilisés dans les analyseurs XML 39  
 format des données renvoyé par FileMaker Server 70  
 UTF-8 encoding 70  
 utilisateurs Web  
 accès à des bases de données protégées 14  
 utilisation des données d'une rubrique Conteneur 18  
 utilisateurs web  
 configuration requise pour l'accès aux solutions de Publication Web personnalisée 23

## V

valeur existante, validation 89  
 valeur unique, validation 89  
 validate(), méthode 90  
 validation  
 année à quatre chiffres 89  
 commandes 88  
 date 89  
 enregistrements 90  
 heure 89  
 heure du jour 89  
 horodatage 89  
 nombre maximal de caractères 88  
 non vide 88  
 numérique uniquement 88  
 rubriques 90  
 version PHP 65  
 -view, commande de requête 52

## W

web\_server\_module\_log.txt, fichier journal 98

## X

### XML

activation dans une base de données 13  
 analyseurs 26, 39  
 avantages 11  
 chaînes de requête 39, 43  
 codage au format UTF-8 29, 39  
 codage de texte URL 29  
 définitions de type de document (DTD) 30, 31, 34  
 dépannage de l'accès aux documents XML 42  
 espaces de nom 30  
 génération données XML à partir d'une requête 25  
 grammaire FMPXMLLAYOUT 36  
 grammaire FMPXMLRESULT 34  
 grammaire fmresultset 31  
 élément 31, 32, 32, 32, 32  
 grammaires comparées 29  
 ordre de traitement de requête 42  
 Publication Web personnalisée, décrite 11  
 récapitulatif des étapes d'accès aux données XML 26  
 réponse, passage d'un modèle à un autre 41  
 requête, spécification d'un modèle 41  
 requêtes pour des données 26  
 spécification XML 1.0 24  
 <xsl:stylesheet> élément 95  
 <xsl:template> élément 95