

FileMaker® Server 18

Custom Web Publishing Handbuch



FileMaker®
An Apple Subsidiary

© 2004-2019 FileMaker, Inc. Alle Rechte vorbehalten.

FileMaker, Inc.
5201 Patrick Henry Drive
Santa Clara, California 95054, USA

FileMaker, FileMaker Cloud, FileMaker Go und das Dateiodner-Logo sind Marken von FileMaker, Inc., eingetragen in den USA und anderen Ländern. FileMaker WebDirect ist eine Marke von FileMaker, Inc. Alle anderen Marken sind Eigentum der jeweiligen Besitzer.

Die FileMaker-Dokumentation ist urheberrechtlich geschützt. Sie dürfen diese Dokumentation ohne schriftliche Genehmigung von FileMaker weder vervielfältigen noch verteilen. Diese Dokumentation darf ausschließlich mit einer gültigen, lizenzierten Kopie der FileMaker-Software verwendet werden.

Alle in den Beispielen erwähnten Personen, Firmen, E-Mail-Adressen und URLs sind rein fiktiv und jegliche Ähnlichkeit mit bestehenden Personen, Firmen, E-Mail-Adressen und URLs ist rein zufällig. Produktdanksagungen sind in den Acknowledgments-Dokumenten aufgeführt, die mit dieser Software geliefert werden. Dokumentdanksagungen sind in den [Acknowledgments-Dokumenten für Dokumentation](#) aufgeführt. Die Erwähnung von Produkten und URLs Dritter dient nur zur Information und stellt keine Empfehlung dar. FileMaker, Inc. übernimmt keine Verantwortung für die Leistung dieser Produkte.

Weitere Informationen finden Sie auf unserer Website unter www.filemaker.com/de/.

Edition: 01

Inhalt

Vorwort	8
Über dieses Handbuch	8
Speicherort der FileMaker-Dokumentation	8
Kapitel 1	
<i>Einführung in Custom Web Publishing</i>	9
Über die Web Publishing Engine	10
Verarbeitung einer Web Publishing Engine -Anforderung	10
Custom Web Publishing mit XML	11
Custom Web Publishing mit PHP	11
XML im Vergleich zu PHP	12
Gründe für XML	12
Gründe für PHP	12
Kapitel 2	
<i>Vorbereiten von Datenbanken für Custom Web Publishing</i>	13
Aktivieren von Custom Web Publishing in einer Datenbank	13
Zugriff auf eine geschützte Datenbank	13
Schützen veröffentlichter Datenbanken	14
Web Server-Unterstützung für Internet-Medientypen (MIME)	16
Veröffentlichen des Inhalts von Containerfeldern im Web	16
In eine Datenbank eingebettete Containerfeldobjekte	16
Containerfelder mit gespeicherten Dateiverweisen	16
Containerfelder mit extern gespeicherten Daten	17
Containerfelder und progressive Downloads	18
Anzeige von Containerfelddaten durch Webbenutzer	18
FileMaker-Scripts und Custom Web Publishing	18
Tipps und Überlegungen zu Scripts	19
Scriptverhalten in Custom Web Publishing-Lösungen	20
Script-Trigger in Custom Web Publishing-Lösungen	20
Kapitel 3	
<i>Custom Web Publishing mit XML</i>	21
Erstellen dynamischer Webseiten mit der Web Publishing Engine	21
Wichtige Funktionen in Custom Web Publishing mit XML	22
Voraussetzungen für Web Publishing	22
Voraussetzungen für das Veröffentlichen einer Datenbank mit Custom Web Publishing	22
Voraussetzungen für Webbenutzer zum Zugriff auf eine Custom Web Publishing-Lösung	23
Verbindung zum Internet oder zu einem Intranet	23
Die nächsten Schritte	23

Kapitel 4

Zugriff auf XML-Daten mit der Web Publishing Engine	24
Verwenden von Custom Web Publishing mit XML	24
Vergleich der Web Publishing Engine mit XML-Import und -Export	24
Wie die Web Publishing Engine aus einer Abfrage XML-Daten generiert	25
Allgemeines Verfahren beim Zugriff auf XML-Daten von der Web Publishing Engine	26
Erläuterung der URL-Syntax für XML-Daten und Containerobjekte	27
Erläuterung der URL-Syntax für XML-Daten	27
Erläuterung der URL-Syntax für FileMaker-Containerobjekte in XML-Lösungen	28
Erläuterung von URL-Textkodierung	29
Zugriff auf XML-Daten über die Web Publishing Engine	29
Erläuterung von Namensräumen für FileMaker XML	30
Erläuterung der FileMaker-Datenbankfehlercodes	30
Abrufen der Document Type Definitions für die FileMaker-Grammatiken	30
Verwenden der fmresultset-Grammatik	30
Beschreibung der Elemente der fmresultset-Grammatik	31
XML-Daten in der fmresultset-Grammatik	33
Verwenden anderer FileMaker XML-Grammatiken	34
Beschreibung der Elemente in der FMPXMLRESULT-Grammatik	34
XML-Daten in der FMPXMLRESULT-Grammatik	35
Beschreibung der Elemente der Grammatik FMPXMLLAYOUT	36
XML-Daten in der FMPXMLLAYOUT-Grammatik	38
Erläuterung von UTF-8-kodierten Daten	39
Abfragen von XML-Daten mithilfe von FileMaker-Abfrage-Strings	39
Wechseln des Layouts für eine XML-Antwort	41
Erläuterung der Verarbeitung einer XML-Abfrage	42
Fehlerbehebung beim XML-Dokumentzugriff	42

Kapitel 5

Gültige Namen in XML-Abfrage-Strings	43
Erläuterung von Abfrage-Befehlen und -Parametern	43
Richtlinien für die Verwendung von Abfrage-Befehlen und -Parametern	44
Abfrage-Befehls-Parsing	45
Erläuterung der Syntax für einen voll qualifizierten Feldnamen	46
Verwenden von Abfrage-Befehlen für Ausschnittfelder	46
Erläuterung der Syntax für das Angeben eines Variablenfelds	48
Abfrage-Befehlsreferenz	49
Abfrage-Befehl –dbnames (Datenbanknamen)	49
Abfrage-Befehl –delete (Datensatz löschen)	49
Abfrage-Befehl –dup (Datensatz duplizieren)	49
Abfrage-Befehl –edit (Datensatz bearbeiten)	49
Abfrage-Befehle –find, –findall oder –findany (Datensätze suchen)	50
Abfrage-Befehl –findquery (Compound-Suche)	50
Abfrage-Befehl –layoutnames (Layoutnamen)	51
Abfrage-Befehl –new (Neuer Datensatz)	51
Abfrage-Befehl –scriptnames (Scriptnamen)	52
Abfrage-Befehl –view (Layoutinformationen anzeigen)	52

Abfrage-Parameterreferenz	52
Abfrage-Parameter –db (Datenbankname)	52
–delete.related (Löschen von Ausschnittdatensätzen, Abfrage-Parameter)	53
Abfrage-Parameter –field (Containerfeldname)	53
Abfrage-Parameter „fieldname“ (Nicht-Containerfeldname)	53
Abfrage-Parameter „fieldname.op“ (Vergleichsoperator)	54
Abfrage-Parameter –lay (Layout)	55
Abfrage-Parameter –lay.response (Layout wechseln für Antwort)	55
Abfrage-Parameter –lop (Logischer Operator)	56
Abfrage-Parameter –max (Maximum Datensätze)	56
Abfrage-Parameter –modid (ÄnderungsID)	57
Abfrage-Parameter –query (Compound-Suchabfrage)	57
Abfrage-Parameter –recid (DatensatzID)	58
Abfrage-Parameter –relatedsets.filter (Ausschnittdatensätze filtern)	59
Abfrage-Parameter –relatedsets.max (Ausschnittdatensätze beschränken)	60
Abfrage-Parameter –script (Script)	60
Abfrage-Parameter –script.param (Parameter an Script übergeben)	61
Abfrage-Parameter –script.prefind (Script vor Suchen)	61
Abfrage-Parameter –script.prefind.param (Parameter an Script vor Suchen übergeben)	62
Abfrage-Parameter –script.presort (Script vor Sortieren)	62
Abfrage-Parameter –script.presort.param (Parameter an Script vor Sortieren übergeben)	62
Abfrage-Parameter –skip (Datensätze überspringen)	63
Abfrage-Parameter –sortfield (Sortierfeld)	63
Abfrage-Parameter –sortorder (Sortierfolge)	64

Kapitel 6

Erläuterung von Custom Web Publishing mit PHP	65
Wichtige Funktionen in Custom Web Publishing mit PHP	65
Anforderungen für Custom Web Publishing	65
Voraussetzungen für das Veröffentlichen einer Datenbank mit Custom Web Publishing	66
Voraussetzungen für Web-Benutzer zum Zugriff auf eine Custom Web Publishing-Lösung	66
Verbindung zum Internet oder zu einem Intranet	66
Manuelle Installation von FileMaker API für PHP	67
Die nächsten Schritte	68

Kapitel 7

Übersicht über Custom Web Publishing mit PHP	69
Funktionsweise der Web Publishing Engine mit PHP-Lösungen	69
Allgemeine Schritte bei Custom Web Publishing mit PHP	69

Kapitel 8

Verwendung von FileMaker API für PHP	72
Weitere Informationen	72
Referenz zu FileMaker API für PHP	72
FileMaker API für PHP-Support	72

Verwendung der Klasse FileMaker	73
FileMaker, Objekte der Klasse	73
FileMaker-Befehlsobjekte	73
Entschlüsseln von Daten für die Verwendung im FileMaker API	74
Herstellen einer Verbindung zu einer FileMaker-Datenbank	75
Arbeiten mit Datensätzen	75
Erstellung eines Datensatzes	75
Duplizieren von Datensätzen	76
Bearbeiten von Datensätzen	76
Löschen von Datensätzen	77
Ausführen von FileMaker-Scripts	77
Abruf der Liste verfügbarer Scripts	77
Ausführen von FileMaker-Scripts	78
Ausführung eines Scripts vor der Ausführung eines Befehls	78
Ausführung eines Scripts vor der Sortierung einer Ergebnismenge	78
Ausführung eines Scripts nach der Erstellung einer Ergebnismenge	79
Ausführungsreihenfolge von Scripts	79
Arbeiten mit FileMaker-Layouts	80
Verwenden von Ausschnitten	81
Auflistung der in einem bestimmten Layout definierten Ausschnitte	81
Abrufen von Ausschnittnamen für ein bestimmtes Ergebnisobjekt	81
Abruf von Informationen über Ausschnitte für ein bestimmtes Layout	81
Abrufen von Informationen für einen bestimmten Ausschnitt	81
Abrufen des Tabellennamens für einen Ausschnitt	82
Abrufen der Ausschnittdatensätze für einen bestimmten Datensatz	82
Erstellen eines neuen Datensatzes in einem Ausschnitt	82
Löschen eines Datensatzes aus einem Ausschnitt	82
Verwenden von Wertelisten	83
Abrufen der Namen aller Wertelisten für ein bestimmtes Layout	83
Abrufen eines Arrays aller Wertelisten für ein bestimmtes Layout	83
Beziehen von Werten für eine benannte Werteliste	83
Ausführen von Suchabfragen	84
Der Befehl „Find All“	85
Der Befehl „Find Any“	85
Der Befehl „Find“	85
Der Befehl „Compound Find“	86
Verarbeitung der Datensätze in einer Ergebnismenge	88
Begrenzen der Ausschnittzeilen, die von Suchabfragen zurückgegeben werden	89
Vorabüberprüfung von Befehlen, Datensätzen und Feldern	89
Vorabüberprüfung von Datensätzen in einem Befehl	91
Vorabüberprüfung von Datensätzen	91
Vorabüberprüfung von Feldern	91
Bearbeitung der Überprüfungsfehler	91
Fehlerbehandlung	93
Kapitel 9	
<i>Einstellen, Testen und Überwachen einer Site</i>	94
Einstellen einer Custom Web Publishing-Site	94

Testen einer Custom Web Publishing-Site	95
Stylesheets zum Testen der XML-Ausgabe	96
Überwachen Ihrer Website	97
Verwenden der Web-Server-Zugriffs- und Fehlerprotokolle	97
Verwenden des Web Publishing Engine-Protokolls	97
Verwenden des Web-Server-Modul-Fehlerprotokolls	99
Verwenden der Tomcat-Protokolle	99
Anhang A	
<i>Fehlercodes für Custom Web Publishing</i>	100
Fehlercodenummern in XML-Format	100
Fehlercodenummern für FileMaker-Datenbanken	101
<i>Index</i>	102

Vorwort

Über dieses Handbuch

Dieses Handbuch setzt voraus, dass Sie mit der Verwendung von FileMaker® Pro Advanced zur Erstellung von Datenbanken vertraut sind. Sie sollten sich mit den Grundlagen der Datenbankgestaltung von FileMaker Pro Advanced auskennen und die Konzepte von Feldern, Beziehungen, Layouts, Ausschnitten und Containerfeldern verstehen. Weitere Informationen zu FileMaker Pro Advanced finden Sie in der [FileMaker Pro Advanced Hilfe](#).

Dieses Handbuch setzt zudem Erfahrung mit der Entwicklung von Websites, speziell Technologien wie XML bzw. PHP für die Integration von FileMaker-Daten in Websites und Web-Anwendungen, voraus.

Dieses Handbuch enthält die folgenden Informationen über Custom Web Publishing mit FileMaker Server:

- Voraussetzungen für die Entwicklung einer Custom Web Publishing-Lösung
- Veröffentlichen Ihrer Datenbanken mit XML
- Abruf von XML-Daten aus Datenbanken, die FileMaker Server bereitstellt
- Veröffentlichen Ihrer Datenbanken mit PHP
- Abruf von Daten aus von FileMaker Server bereitgestellten Datenbanken mithilfe des FileMaker API für PHP
- Voraussetzungen für Webbenutzer zum Zugriff auf eine Custom Web Publishing-Lösung

Speicherort der FileMaker-Dokumentation

- Die FileMaker Server Hilfe ist auf jeder Seite von FileMaker Server Admin Console verfügbar. Scrollen Sie zum Ende der Seite und klicken Sie auf **Hilfe**.
- Die Dokumentation für FileMaker Server ist auf jeder Seite von Admin Console abrufbar. Scrollen Sie zum Ende der Seite und klicken Sie auf **Dokumentation**.
- Rufen Sie im Web das [Produktdokumentationszentrum](#) auf.

Kapitel 1

Einführung in Custom Web Publishing

Mit FileMaker Server können Sie Ihre FileMaker-Datenbank im Internet oder in einem Intranet wie folgt veröffentlichen:

FileMaker WebDirect: Mit FileMaker WebDirect können Sie Layouts aus einer Datenbank schnell und einfach im Web veröffentlichen. Sie müssen keine zusätzliche Software installieren – mit einem kompatiblen Webbrowser und Zugang zum Internet bzw. einem Intranet können Webbenutzer eine Verbindung zu Ihrer FileMaker WebDirect-Lösung herstellen, um Datensätze anzuzeigen, zu bearbeiten, zu sortieren oder zu durchsuchen, wenn Sie ihnen die entsprechenden Zugriffsrechte geben.

Für FileMaker WebDirect muss der Hostcomputer FileMaker Server ausführen. Die Benutzeroberfläche gleicht der FileMaker Pro Advanced-Anwendung. Die Webseiten und Formulare, mit denen der Webbenutzer arbeitet, hängen von den in der FileMaker Pro Advanced-Datenbank definierten Layouts und Ansichten ab. Weitere Informationen finden Sie im Handbuch [FileMaker WebDirect](#).

Statisches Veröffentlichen: Wenn sich Ihre Daten selten ändern oder wenn Sie keine Live-Verbindung Ihrer Benutzer zu Ihrer Datenbank wünschen, können Sie die Daten statisch veröffentlichen. Bei der statischen Veröffentlichung exportieren Sie Daten aus einer FileMaker Pro Advanced-Datenbank, um eine Webseite zu erstellen, die Sie anhand von HTML weiter anpassen können. Die Webseite ändert sich nicht, wenn sich Informationen in Ihrer Datenbank ändern, und die Benutzer stellen keine Verbindung zu Ihrer Datenbank her. (Mit FileMaker WebDirect werden die Daten im Webbrowser immer dann aktualisiert, wenn die Daten in der Datenbank aktualisiert werden.) Siehe [FileMaker Pro Advanced Hilfe](#).

FileMaker Data API: Wenn Sie mit der Representational State Transfer- (REST-) Architektur vertraut sind: FileMaker stellt eine REST API-Implementierung bereit, die es Web-Diensten gestattet, auf Daten in bereitgestellten Lösungen zuzugreifen. Ihr Webdienst ruft das FileMaker Data API auf, um einen Authentifizierungstoken für den Zugriff auf eine bereitgestellte Lösung zu erhalten, und verwendet dann in nachfolgenden Aufrufen diesen Token, um Datensätze zu erstellen, zu aktualisieren und zu löschen sowie Suchabfragen durchzuführen. Das FileMaker Data API gibt Daten in JavaScript Object Notation (JSON) zurück. Informationen hierzu finden Sie im Handbuch [FileMaker Data API](#).

Custom Web Publishing: Um Ihre FileMaker-Datenbank in eine angepasste Website zu integrieren, verwenden Sie die Techniken für Custom Web Publishing, die in FileMaker Server zur Verfügung stehen. Für FileMaker Server, der die veröffentlichten Datenbanken bereitstellt, muss FileMaker Pro Advanced weder installiert noch gestartet sein, um Custom Web Publishing verfügbar zu machen.

Mit Custom Web Publishing können Sie:

- Ihre Datenbank in andere Websites integrieren.
- festlegen, wie Benutzer mit den Daten arbeiten.
- steuern, wie Daten in Webbrowsern angezeigt werden.

FileMaker Server stellt zwei Custom Web Publishing-Technologien zur Verfügung:

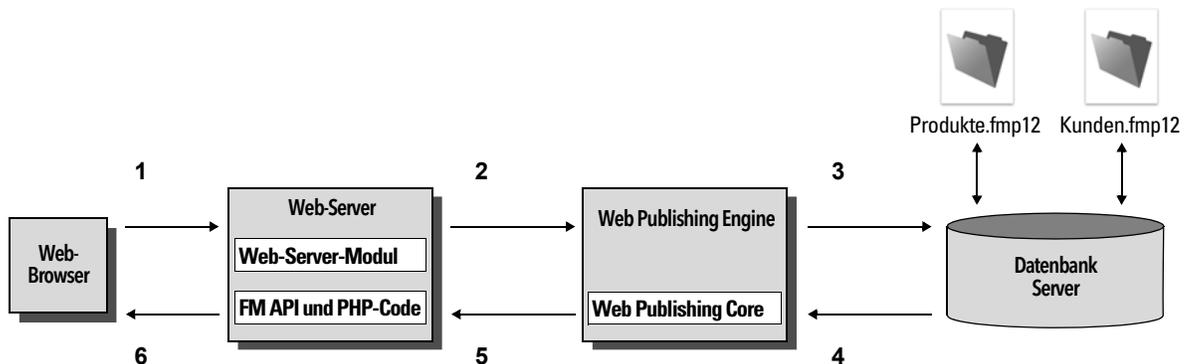
- Custom Web Publishing mit XML: Verwenden Sie XML Data Publishing, um FileMaker-Daten mit anderen Webseiten und Anwendungen auszutauschen. Indem Sie HTTP-URL-Abfragen mit FileMaker-Abfrage-Befehlen und -Parametern verwenden, können Sie eine Datenbank abfragen, die von FileMaker Server bereitgestellt wird, die resultierenden Daten in XML-Format herunterladen und die resultierenden XML-Daten so verwenden, wie Sie es möchten.
- Custom Web Publishing mit PHP: Verwenden Sie das FileMaker API für PHP, das eine objektorientierte PHP-Schnittstelle zu FileMaker Pro Advanced-Datenbanken bereitstellt, um Ihre FileMaker-Daten in einer PHP-Web-Anwendung zu integrieren. Da Sie die PHP-Webseiten selbst programmieren, haben Sie die komplette Kontrolle über die Benutzeroberfläche und darüber, wie der Benutzer mit den Daten arbeitet.

Über die Web Publishing Engine

Um FileMaker WebDirect und Custom Web Publishing zu unterstützen, verwendet FileMaker Server einen Satz von Softwarekomponenten, die sogenannte *FileMaker Server Web Publishing Engine*. Die Web Publishing Engine verwaltet Interaktionen zwischen dem Browser eines Webbenutzers, Ihrem Web-Server und FileMaker Server.

Custom Web Publishing mit XML: Webbenutzer greifen auf Ihre Custom Web Publishing-Lösung zu, indem sie auf einen HREF-Link klicken oder einen Uniform Resource Locator (URL) eingeben, der die Web-Server-Adresse und eine FileMaker-Abfrage-Zeichenfolgenabfrage angibt. Die Web Publishing Engine gibt die in der Abfrage-Zeichenfolgenabfrage angegebenen XML-Daten zurück.

Custom Web Publishing mit PHP: Wenn ein Webbenutzer auf Ihre Custom Web Publishing-Lösung zugreift, stellt PHP auf FileMaker Server eine Verbindung mit der Web Publishing Engine her und reagiert über das FileMaker API für PHP.



Verwenden der FileMaker Server Web Publishing Engine für Custom Web Publishing

Verarbeitung einer Web Publishing Engine -Anforderung

1. Eine Anforderung wird von einem Webbrowser oder Programm an den Web-Server gesendet.
2. Der Web-Server leitet die Anforderung über das FileMaker Web-Server-Modul an die Web Publishing Engine weiter.
3. Die Web Publishing Engine fordert Daten von der Datenbank an, die der Datenbank-Server bereitstellt.

4. FileMaker Server sendet die angeforderten FileMaker-Daten an die Web Publishing Engine.
5. Die Web Publishing Engine konvertiert die FileMaker-Daten, um die Anforderung zu beantworten.
 - PHP-Abfragen konvertiert das FileMaker API für PHP in entsprechende XML-Abfragen. Die Web Publishing Engine verarbeitet die XML-Abfrage und sendet XML-Daten zurück an das FileMaker API für PHP. Das FileMaker API für PHP konvertiert dann die XML-Daten in PHP-Objekte, die die PHP-Anwendung nutzen kann.
 - Für XML-Anforderungen sendet die Web Publishing Engine die XML-Daten direkt an den Web-Server.
6. Der Web-Server sendet die Ausgabe an den anfordernden Webbrowser oder das anfordernde Programm.

Wichtig Sicherheitsüberlegungen sind wichtig, wenn Sie Daten im Web veröffentlichen. Beachten Sie die Sicherheitsrichtlinien im [FileMaker Sicherheitshandbuch](#).

Informationen über das Einrichten von Datenbanken für Custom Web Publishing finden Sie in Kapitel 2 „Vorbereiten von Datenbanken für Custom Web Publishing“.

Custom Web Publishing mit XML

FileMaker Custom Web Publishing mit XML ermöglicht Ihnen, Abfrageanforderungen an eine FileMaker Pro Advanced-Datenbank zu senden, die von FileMaker Server bereitgestellt wird, und die resultierenden Daten anzuzeigen und zu ändern. Mithilfe einer HTTP-Abfrage mit den geeigneten Abfrage-Befehlen und -Parametern können Sie FileMaker-Daten als ein XML-Dokument abrufen. Sie können die XML-Daten in andere Anwendungen exportieren.

Custom Web Publishing mit PHP

Das FileMaker API für PHP bietet eine objektorientierte PHP-Schnittstelle zu FileMaker-Datenbanken. Mithilfe des FileMaker API für PHP kann sowohl auf Daten als auch auf Logik, die in einer FileMaker Pro Advanced-Datenbank gespeichert sind, zugegriffen werden. Diese können auch im Internet veröffentlicht oder an andere Anwendungen exportiert werden. Das API unterstützt zudem komplexe und zusammengesetzte Suchabfragen für das Extrahieren und Filtern von in FileMaker Pro Advanced-Datenbanken gespeicherten Daten.

Ursprünglich als prozedurale Programmiersprache entwickelt, wurde PHP als objektorientierte Programmiersprache für die Web-Entwicklung ausgebaut. PHP bietet Programmiersprachenfunktionalität für den Aufbau praktisch jeder Art von Logik innerhalb einer Site-Seite. Zum Beispiel können Sie bedingte Logikstrukturen verwenden, um die Seitengenerierung, das Datenrouting oder den Workflow zu steuern. PHP bietet zudem Funktionen für Site-Administration und Sicherheit.

XML im Vergleich zu PHP

Der folgende Abschnitt zeigt einige Richtlinien auf, um die beste Lösung für Ihre Site zu bestimmen.

Gründe für XML

- Die Syntax der FileMaker XML-Abfrageparameter ist für die Datenbankinteraktion konzipiert und vereinfacht die Lösungsentwicklung.
- XML ist ein W3C-Standard.
- XML ist ein von Maschinen und Menschen lesbares Format, das Unicode unterstützt, so dass Daten in jeder geschriebenen Sprache kommuniziert werden können.
- XML ist bestens geeignet für die Darstellung von Datensätzen, Listen und Daten mit Baumstruktur.
- Sie können FMPXMLRESULT für den Zugriff auf XML-Daten mit Custom Web Publishing und für den XML-Export aus FileMaker Pro Advanced-Datenbanken verwenden.

Hinweis Weitere Informationen über Custom Web Publishing mit XML finden Sie in Kapitel 3 „Custom Web Publishing mit XML“.

Gründe für PHP

- PHP ist eine leistungsstarke, objektorientierte prozedurale Scriptingsprache und relativ leicht zu erlernen. Es gibt viele Ressourcen für Schulung, Entwicklung und Support.
- Mithilfe des FileMaker API für PHP kann auf Daten und Logik, die in einer FileMaker Pro Advanced-Datenbank gespeichert sind, zugegriffen werden. Diese können auch im Internet veröffentlicht oder an andere Anwendungen exportiert werden.
- Mit PHP können Sie bedingte Logik verwenden, um den Seitenaufbau oder den Fluss zu kontrollieren.
- PHP bietet Programmiersprachenfunktionalität für den Aufbau vieler Arten von Logik innerhalb einer Site-Seite.
- PHP ist eine der beliebtesten Web-Scriptingsprachen.
- PHP ist eine Open-Source-Sprache, verfügbar unter php.net.
- PHP ermöglicht den Zugriff auf eine große Vielzahl von Third-Party-Komponenten, die Sie in Ihre Lösungen integrieren können.

Hinweis Weitere Informationen über Custom Web Publishing mit PHP finden Sie in Kapitel 6 „Erläuterung von Custom Web Publishing mit PHP“.

Kapitel 2

Vorbereiten von Datenbanken für Custom Web Publishing

Bevor Sie Custom Web Publishing mit einer Datenbank verwenden können, müssen Sie die Datenbank vorbereiten und vor unbefugtem Zugriff schützen.

Aktivieren von Custom Web Publishing in einer Datenbank

Sie müssen ein erweitertes Zugriffsrecht „Custom Web Publishing“ in jeder Datenbank aktivieren, die Sie veröffentlichen wollen. Wenn Sie ein erweitertes Zugriffsrecht „Custom Web Publishing“ in der Datenbank nicht aktivieren, können Webbenutzer Custom Web Publishing nicht nutzen, um auf die Datenbank zuzugreifen, selbst wenn sie von FileMaker Server bereitgestellt ist, der für eine Web Publishing Engine konfiguriert wurde.

So aktivieren Sie Custom Web Publishing in einer Datenbank:

1. Öffnen Sie in FileMaker Pro Advanced die zu veröffentlichende Datenbank mit einem Konto, das über die Berechtigung für vollen Zugriff verfügt. Alternativ können Sie die Datenbank mit einem Konto öffnen, das über die Berechtigung „Erweiterte Zugriffsrechte verwalten“ verfügt.
2. Weisen Sie das erweiterte Zugriffsrecht „Custom Web Publishing“ zu, das Sie verwenden möchten:
 - Verwenden Sie für Custom Web Publishing mit XML fmxml.
 - Verwenden Sie für Custom Web Publishing mit PHP fmphp.
3. Weisen Sie eine Berechtigung, die das erweiterte Custom Web Publishing-Zugriffsrecht enthält, einem oder mehreren Konten bzw. dem Admin- oder Gastkonto zu.

Hinweis Wenn Sie Kontonamen und Passwörter für Custom Web Publishing-Lösungen definieren, verwenden Sie druckbare ASCII-Zeichen. Beispiel: **a-z**, **A-Z** und **0-9**. Verwenden Sie für sicherere Kontonamen und Passwörter zudem Satzzeichen wie „!“ und „%“, aber verwenden Sie keine Doppelpunkte. Informationen über das Einrichten von Konten finden Sie in der [FileMaker Pro Advanced Hilfe](#).

Zugriff auf eine geschützte Datenbank

Custom Web Publishing ermöglicht Ihnen, den Zugriff auf Ihre veröffentlichten Datenbanken durch Passwortschutz, Datenbankverschlüsselung und sichere Verbindungen einzuschränken. Wenn Webbenutzer eine Custom Web Publishing-Lösung für den Zugriff auf eine Datenbank verwenden, werden sie eventuell aufgefordert, ihre Kontoinformationen einzugeben. Wenn das Gastkonto für die Datenbank deaktiviert ist oder über keine Berechtigung mit dem erweiterten Zugriffsrecht für Custom Web Publishing verfügt, verwendet die Web Publishing Engine die HTTP-Standardauthentifizierung, um entsprechende Informationen von Webbenutzern zu verlangen. Der Browser des Webbenutzers zeigt das Dialogfeld für die HTTP-Standardauthentifizierung an, in dem der Benutzer einen Benutzernamen und ein Passwort für ein Konto eingibt, das über ein erweitertes Zugriffsrecht für Custom Web Publishing verfügt.

Die folgende Liste fasst die Abläufe zusammen, wenn ein Webbenutzer eine Custom Web Publishing-Lösung für den Zugriff auf eine Datenbank verwendet:

- Wenn Sie einem Konto kein Passwort zugewiesen haben, geben Webbenutzer nur den Kontonamen an.
- Wenn das Gastkonto deaktiviert ist, werden die Benutzer beim Zugriff auf die Datenbank aufgefordert, Kontoname und Passwort anzugeben. Für das Konto muss ein erweitertes Zugriffsrecht für Custom Web Publishing aktiviert sein.
- Wenn das Gastkonto aktiviert ist und über eine Berechtigung mit einem erweiterten Zugriffsrecht für Custom Web Publishing verfügt, öffnen automatisch alle Webbenutzer die Datenbank mit den Zugriffsrechten, die dem Gastkonto zugewiesen sind. Wenn das erweiterte Zugriffsrecht für Custom Web Publishing dem Gastkonto zugewiesen ist:
 - Webbenutzer werden nicht aufgefordert, Kontoname und Passwort einzugeben, wenn sie eine Datei öffnen.
 - Alle Webbenutzer werden automatisch mit dem Gastkonto angemeldet und erhalten die Gast-Zugriffsrechte. Sie können Benutzern über den Scriptschritt „Erneut anmelden“ erlauben, ihre Anmeldekonto von einem Webbrowser aus zu wechseln (z. B. vom Gastkonto zu einem anderen Konto mit mehr Rechten).
 - Die Standardberechtigungen für Gastkonten umfassen den Nur-Lese-Zugriff. Sie können für dieses Konto die Standardberechtigungen einschließlich erweiterten Zugriffsrechten ändern. Siehe [FileMaker Pro Advanced Hilfe](#).
- Wenn ein Webbenutzer gültige Kontodaten eingegeben hat, werden diese Kontodaten so lange verwendet, bis eine Zeitüberschreitung der Browser-Sitzung auftritt. Bei einer Zeitüberschreitung der Browser-Sitzung wird der Webbenutzer aufgefordert, erneut ein gültiges Konto anzugeben.

Hinweis Standardmäßig können Webbenutzer ihr Kontopasswort nicht von einem Webbrowser aus ändern. Sie können diese Funktion über den Scriptschritt „Passwort ändern“ in Ihre Datenbank integrieren, damit Webbenutzer ihr Passwort vom Browser aus ändern können. Siehe [FileMaker Pro Advanced Hilfe](#).

Schützen veröffentlichter Datenbanken

Wenn Sie Custom Web Publishing verwenden, können Sie beschränken, wer auf Ihre veröffentlichten Datenbanken zugreifen kann.

- Weisen Sie Datenbankkonten, die für Custom Web Publishing verwendet werden, Passwörter zu.
- Aktivieren Sie das erweiterte Zugriffsrecht „Custom Web Publishing“ nur in den Berechtigungen für Konten, denen Sie den Zugriff auf Ihre veröffentlichten Datenbanken erlauben wollen.
- Deaktivieren Sie das erweiterte Zugriffsrecht „Custom Web Publishing“ für bestimmte Datenbanken, indem Sie das erweiterte Zugriffsrecht „fmxml“ oder „fmphp“ für sämtliche Berechtigungen in dieser Datenbank deaktivieren. Siehe [FileMaker Pro Advanced Hilfe](#).

- Aktivieren oder deaktivieren Sie Custom Web Publishing für alle Custom Web Publishing-Lösungen, die die Befehlszeilenschnittstelle (CLI) verwenden. Verwenden Sie zusätzlich FileMaker Server Admin Console, um die Web Publishing Engine zu starten bzw. zu stoppen. Weitere Informationen finden Sie in der [FileMaker Server Hilfe](#).
- Konfigurieren Sie Ihren Web-Server, um die IP-Adressen einzuschränken, die über die Web Publishing Engine auf Ihre Datenbanken zugreifen können. Beispielsweise können Sie angeben, dass nur Webbenutzer von der IP-Adresse 192.168.100.101 auf Ihre Datenbanken zugreifen können. Informationen über das Einschränken von IP-Adressen finden Sie in der Dokumentation zu Ihrem Web-Server.

FileMaker Server unterstützt Verschlüsselung für Daten, die auf Platte geschrieben werden, und für Daten, die an Clients übertragen werden.

- Verschlüsseln Sie Ihre Datenbank mithilfe der Funktion „Datenbankverschlüsselung“ von FileMaker Pro Advanced. Verschlüsselung schützt die FileMaker-Datenbankdatei und alle temporären Dateien, die auf Platte geschrieben werden. Informationen hierzu finden Sie im Handbuch [FileMaker Server Installation und Konfiguration](#) und in der [FileMaker Pro Advanced Hilfe](#).
 - Eine verschlüsselte Datenbank, die auf FileMaker Server bereitgestellt wird, wird über Admin Console oder die CLI geöffnet. Als FileMaker Server-Administrator öffnen Sie die Datei mit dem Datenbank-Verschlüsselungspasswort, damit FileMaker-Clients die verschlüsselte Datenbank nutzen können.
 - Sobald der FileMaker Server-Administrator die verschlüsselte FileMaker-Datenbank mit dem Verschlüsselungspasswort geöffnet hat, brauchen FileMaker-Clients das Verschlüsselungspasswort nicht mehr, um auf die verschlüsselte Datenbank zuzugreifen. Weitere Informationen zum Öffnen einer verschlüsselten Datenbank finden Sie in der [FileMaker Server Hilfe](#).
- Verwenden Sie Secure Sockets Layer- (SSL-) Verschlüsselung für die Kommunikation zwischen Ihrem Web-Server und den Webbrowsern. Auf SSL-Verbindungen erfolgt der Zugriff über eine HTTPS-Verbindung. FileMaker Server bietet ein von FileMaker, Inc. signiertes Standard-SSL-Zertifikat an, das den Servernamen nicht prüft. Das Standardzertifikat von FileMaker dient nur für Testzwecke. Ein eigenes SSL-Zertifikat wird für die Produktion benötigt. Weitere Informationen finden Sie im Handbuch [FileMaker Server Installation und Konfiguration](#).
Wenn Sie ein eigenes SSL-Zertifikat importieren, verwenden Datenbank-Server-Client-Verbindungen SSL und HTTP-Verbindungen werden über HTTPS geroutet. Verwenden Sie mit einem eigenen SSL-Zertifikat das HTTPS-Verzeichnis für die Bereitstellung von PHP-Site-Dateien. Siehe Kapitel 7 „Allgemeine Schritte bei Custom Web Publishing mit PHP“.

Weitere Informationen über die Sicherheit Ihrer Datenbank finden Sie im [FileMaker Sicherheitshandbuch](#).

Hinweis Aus Sicherheitsgründen verwenden Webseiten, die von anderen Web-Servern bereitgestellt werden, möglicherweise keine <iframe>-Tags, um Custom Web Publishing-Inhalte einzubetten. Damit Sie Custom Web Publishing-Inhalte in die <iframe>-Tags von separaten Webseiten einbetten können, müssen diese Webseiten vom FileMaker Server-Web-Server bereitgestellt werden.

Web Server-Unterstützung für Internet-Medientypen (MIME)

Ihr Web-Server bestimmt die Unterstützung für die aktuellen MIME-Typen (MIME; Multipurpose Internet Mail Extensions), die für das Internet registriert sind. Die Web Publishing Engine ändert die MIME-Unterstützung eines Web-Servers nicht. Weitere Informationen finden Sie in der Dokumentation zu Ihrem Web-Server.

Veröffentlichen des Inhalts von Containerfeldern im Web

Der Inhalt eines Containerfelds kann in die Datenbank eingebettet sein, per Verweis über einen relativen Pfad verknüpft sein oder extern gespeichert sein.

In eine Datenbank eingebettete Containerfeldobjekte

Wenn ein Containerfeld die tatsächlichen Dateien in der FileMaker-Datenbank speichert, ist keine Aktion mit dem Inhalt des Containerfelds erforderlich, sofern die Datenbankdatei korrekt auf dem FileMaker Server freigegeben und zugänglich ist. Siehe „Erläuterung der URL-Syntax für FileMaker-Containerobjekte in XML-Lösungen“ auf Seite 28.

Containerfelder mit gespeicherten Dateiverweisen

Wenn ein Containerfeld einen Dateiverweis speichert, gehen Sie wie folgt vor, um die Dateien, auf die verwiesen wird, mit der Web Publishing Engine zu veröffentlichen.

1. Speichern Sie die Containerobjektdateien im Web-Ordner des FileMaker Pro Advanced-Ordners.
2. Fügen Sie in FileMaker Pro Advanced die Objekte in das Containerfeld ein und wählen Sie die Option **Nur Verweis auf die Datei speichern**.
3. Kopieren oder verschieben Sie die Objektdateien, auf die verwiesen wird, im Web-Ordner an die gleiche Stelle im relativen Pfad des Root-Ordners der Web-Server-Software.
 - Für IIS (Windows):
[Laufwerk]:\Programme\FileMaker\FileMaker Server\HTTPServer\conf
wobei [Laufwerk] das Laufwerk bezeichnet, auf dem sich die Web Publishing Engine-Komponente Ihres FileMaker Server-Einsatzes befindet.
 - Für Apache (macOS): /Library/FileMaker Server/HTTPServer/htdocs

Hinweis Für Containerobjekte, die als Dateiverweise gespeichert sind, muss Ihr Web-Server so konfiguriert sein, dass er die MIME-Typen für alle Arten von Dateien unterstützt, die Sie anbieten wollen (z. B. Filme). Ihr Web-Server bestimmt die Unterstützung für die aktuellen MIME-Typen, die für das Internet registriert sind. Die Web Publishing Engine ändert die MIME-Unterstützung eines Web-Servers nicht. Weitere Informationen finden Sie in der Dokumentation zu Ihrem Web-Server.

Containerfelder mit extern gespeicherten Daten

Wenn ein Containerfeld Objekte extern speichert – wenn Sie also **Containerdaten extern speichern** in dem FileMaker Pro Advanced-Dialogfeld „Feldoptionen“ ausgewählt haben – verwenden Sie FileMaker Pro Advanced, um Datenbankdateien aus dem Client-Dateisystem zu FileMaker Server zu übertragen. Wenn Sie eine Datenbank mithilfe von FileMaker Pro Advanced hochladen, werden dabei die extern gespeicherten Containerfelddaten auf FileMaker Server hochgeladen. In der [FileMaker Pro Advanced Hilfe](#) finden Sie Informationen zum Übertragen der Datenbankdateien auf FileMaker Server.

Wenn Sie eine Datenbank manuell hochladen, die ein Containerfeld mit extern gespeicherten Objekten verwendet, müssen Sie diese Schritte befolgen, um die extern gespeicherten Containerobjekte über die Web Publishing Engine zu veröffentlichen.

So laden Sie eine Datenbank manuell hoch:

1. Legen Sie die Datenbankdatei an den richtigen Ort auf dem Server. Legen Sie die FileMaker Pro Advanced-Datenbankdateien, die FileMaker Server öffnen soll – oder Verknüpfungen (Windows) bzw. Aliasdateien (macOS) zu diesen Dateien – in den folgenden Ordnern ab:
 - Windows:
`[Laufwerk]:\Programme\FileMaker\FileMaker Server\Data\Databases\`
wobei [Laufwerk] das Hauptlaufwerk ist, von dem das System startet.
 - macOS: `/Library/FileMaker Server/Data/Databases/`Sie können die Dateien auch in einem optional angegebenen, zusätzlichen Datenbankordner ablegen.
2. Erstellen Sie in dem Ordner, in den Sie die Datenbank gelegt haben, einen Ordner namens `RC_Data_FMS`, falls er nicht bereits existiert.
3. Erstellen Sie im Ordner `RC_Data_FMS` einen Ordner mit einem Namen, der dem Namen Ihrer Datenbank entspricht. Wenn Ihre Datenbank z. B. „Kunden“ heißt, erstellen Sie einen Ordner mit dem Namen „Kunden“. Legen Sie die extern gespeicherten Objekte in den neu erstellten Ordner.

Hinweis Wenn Datenbanken auf FileMaker Server bereitgestellt werden, gibt es keine Möglichkeit für verschiedene Datenbanken, auf einen gemeinsamen Ordner mit Containerobjekten zuzugreifen. Die Containerobjekte für jede Datenbank müssen sich in einem Ordner befinden, der durch den Namen der Datenbank identifiziert wird.

4. Für gemeinsam genutzte Dateien von macOS nehmen Sie die Dateien in die Gruppe `fmsadmin` auf.

Weitere Informationen zum manuellen Hochladen von Datenbanken finden Sie in der [FileMaker Server Hilfe](#).

Containerfelder und progressive Downloads

Die Web Publishing Engine unterstützt den progressiven Download von Audiodateien (.mp3), Videodateien (.mov, .mp4 und .avi empfohlen) und PDF-Dateien für interaktive Container. Zum Beispiel kann ein Webbenutzer beginnen, einen Film anzusehen, auch wenn der Film noch nicht komplett heruntergeladen ist. Um den progressiven Download zuzulassen, müssen Sie die Dateien mithilfe von Optionen erstellen, die Streaming unterstützen oder für die Anzeige im Internet optimiert wurden. Erstellen Sie beispielsweise PDF-Dateien mithilfe der Option „Für Web-Anzeige optimieren“.

Wenn Sie ein eigenes SSL-Zertifikat importieren, verwenden Datenbank-Server-Client-Verbindungen SSL und HTTP-Verbindungen werden über HTTPS geroutet. FileMaker Server verwendet sichere Verbindungen, um Daten über HTTPS zu übertragen.

- Interaktive Containerdaten werden über HTTPS heruntergeladen.
- Die Daten sind so sicher als wäre die Lösung eine lokale Datenbank, da keine temporären Cache-Dateien erstellt und die Daten bei der Übertragung verschlüsselt werden.

Wenn Sie kein eigenes SSL-Zertifikat besitzen, werden die Verbindungen, über die FileMaker Server Daten überträgt, während der Übertragung nicht verschlüsselt und die Daten werden über HTTP übertragen.

- FileMaker-Clients sehen die interaktiven Containerdaten mit geringer Verzögerung.
- FileMaker Server speichert die Containerfelddaten in einen Cache-Ordner auf dem Server zwischen, wenn ein FileMaker Pro Advanced-, FileMaker Go- oder Web-Client die Daten anfordert. Die Daten können im Cache-Ordner zwei Stunden lang zwischengespeichert verbleiben, bis FileMaker Server den Cache-Ordner periodisch leert. Die Daten werden nicht lokal auf dem Client im Cache abgelegt.

Anzeige von Containerfelddaten durch Webbenutzer

Bei der Veröffentlichung einer Datenbank mit der Web Publishing Engine gelten folgende Einschränkungen für die Containerfeldobjekte:

- Webbenutzer können den Inhalt eines Containerfelds nicht ändern oder ihm Inhalte hinzufügen. Webbenutzer können Containerfelder nicht verwenden, um Objekte in die Datenbank hochzuladen.
- Bei Datenbanken, die ein Containerfeld mit aktivierter Vorschau verwenden, lädt die Web Publishing Engine die gesamte Datei, keine Vorschau, herunter.

FileMaker-Scripts und Custom Web Publishing

Die Funktion „Scripts verwalten“ in FileMaker Pro Advanced kann sich häufig wiederholende Aufgaben automatisieren und mehrere Aufgaben zusammenfassen. In Kombination mit Custom Web Publishing ermöglichen FileMaker-Scripts den Webbenutzern, mehrere Aufgaben oder eine Reihe von Aufgaben durchzuführen.

FileMaker unterstützt viele 70 Scriptschritte in Custom Web Publishing. Webbenutzer können eine Vielzahl automatisierter Aufgaben durchführen, wenn Sie Scripts in einer Abfrage-Zeichenfolge für einen URL verwenden. Um die Scriptschritte im FileMaker Pro Advanced-Scriptarbeitsbereich anzuzeigen, die Custom Web Publishing unterstützt, klicken Sie auf die Schaltfläche **Kompatibilität** und wählen Sie **Custom Web Publishing**. Scriptschritte, die nicht grau dargestellt werden, werden für Custom Web Publishing unterstützt. Informationen zum Erstellen von Scripts finden Sie in der [FileMaker Pro Advanced Hilfe](#).

Tipps und Überlegungen zu Scripts

Auch wenn viele Scriptschritte im Web identisch funktionieren, gibt es einige, die sich anders verhalten. Weitere Informationen erhalten Sie unter „Scriptverhalten in Custom Web Publishing-Lösungen“ auf Seite 20. Testen Sie alle Scripts, die von einem Webbrowser ausgeführt werden, bevor Sie Ihre Datenbank bereitstellen. Stellen Sie sicher, dass Sie sich mit unterschiedlichen Benutzerkonten anmelden, damit Sie sichergehen können, dass die Scripts für alle Clients gleichermaßen funktionieren. Prüfen Sie die Protokolldatei der Web Publishing Engine (wpe.log) auf Scripting-bezogene Fehler. Siehe „Verwenden des Web Publishing Engine-Protokolls“ auf Seite 97.

Beachten Sie folgende Tipps und Überlegungen:

- Berücksichtigen Sie, welche Werte ein Script zurückgeben kann. Bereiten Sie die Verarbeitung aller zurückgegebenen Daten vor. In FileMaker Pro Advanced kann ein Script alle Datensätze aus einer Tabelle oder der aktuellen Ergebnismenge zurückgeben. Wenn jedoch ein Script alle Datensätze aus einer Tabelle liefert, verfügt eine Webanwendung möglicherweise nicht über genügend Arbeitsspeicher, um die Datensätze zu verarbeiten. Erwägen Sie die Verwendung des Abfrageparameters `-max` in XML-Abfragen bzw. die Methode `setRange()` in PHP-Abfragen, um die Anzahl der zurückgegebenen Datensätze einzuschränken.
- Verwenden Sie Konten und Zugriffsrechte, um die Scripts einzuschränken, die ein Web-Benutzer ausführen kann. Überprüfen Sie, ob die Scripts nur Web-kompatible Scriptschritte enthalten und nur Zugang zu Scripts gewähren, die von einem Webbrowser aus ausgeführt werden sollen.
- Berücksichtigen Sie die Nebeneffekte von Scripts, die eine Reihe von Scriptschritten ausführen, die durch Zugriffsrechte kontrolliert werden. Wenn ein Script beispielsweise einen Scriptschritt zum Löschen von Datensätzen enthält und der Webbenutzer sich nicht mit einem Konto anmeldet, das das Löschen von Datensätzen zulässt, führt das Script den Scriptschritt zum Löschen von Datensätzen nicht aus. Das Script könnte jedoch weiter ausgeführt werden, so dass unerwartete Ergebnisse auftreten können.
- Geben Sie einem Script im Scriptarbeitsbereich volle Zugriffsrechte, damit dieses Script Aufgaben ausführen kann, für die Sie dem Benutzer an sich keinen Zugang gewähren wollen. Beispielsweise können Sie Benutzern untersagen, mit ihren Konten und Zugriffsrechten Datensätze zu löschen, ihnen aber die Ausführung eines Scripts gestatten, das unter vordefinierten Bedingungen innerhalb eines Scripts bestimmte Arten von Datensätzen löscht.
- Damit Scripts Plugins für Custom Web Publishing- und FileMaker WebDirect-Lösungen installieren können, verwenden Sie FileMaker Server Admin Console, um die notwendigen Einstellungen zu aktivieren. Setzen Sie für **Web Publishing-Plugins** die Optionen **Web Publishing-Plugins** und **Scriptschritt „Plugin-Datei installieren“** auf **Aktiviert**. Um Scripts daran zu hindern, Plugins für Web Publishing-Lösungen zu installieren, stellen Sie diese auf **Deaktiviert**.
- Einige Scripts, die mit einem Schritt von einem FileMaker Pro Advanced-Client aus funktionieren, benötigen eventuell einen zusätzlichen Scriptschritt „Schreibe Änderung Datens./Abfrage“, um die Daten auf dem Host zu speichern. Da Webbenutzer nicht über eine direkte Verbindung zum Host verfügen, werden sie nicht benachrichtigt, wenn sich Daten ändern. Funktionen wie bedingte Wertelisten funktionieren z. B. nicht gleichermaßen für Webbenutzer, da die Daten auf dem Host gespeichert werden müssen, bevor sich die Wirkung in einem Wertelistenfeld zeigt.

- Scripts, die Daten ändern, sollten den Scriptschritt „Schreibe Änderung Datens./Abfrage“ enthalten, da Datenänderungen erst im Browser sichtbar werden, wenn die Daten gespeichert bzw. am Server „bestätigt“ wurden. Dies gilt für einige Scriptschritte wie Ausschneiden, Kopieren, Einfügen usw. Viele Aktionen mit nur einem Scriptschritt sollten in Scripts umgewandelt werden, die den Schritt „Schreibe Änderung Datens./Abfrage“ enthalten. Wenn Sie Scripts entwerfen, die von einem Webbrowser aus ausgeführt werden, nehmen Sie den Schritt „Schreibe Änderung Datens./Abfrage“ am Ende des Scripts auf, um sicherzustellen, dass alle Änderungen gespeichert werden.
- Scripts, die von Daten abhängen, die geändert, aber noch nicht bestätigt wurden, funktionieren ggf. anders als in einem FileMaker Pro Advanced-Client. Wenn beispielsweise ein Feld eine Formel „Hole (GeänderteFelder)“ enthält, gibt die Formel „Hole (GeänderteFelder)“ eine leere Liste zurück, wenn der Datensatz über eine XML-Abfrage oder einen PHP-Aufruf bearbeitet wird. Wenn die Custom Web Publishing-Lösung jedoch ein FileMaker-Script ausführt, das den Datensatz bearbeitet, gibt die Formel „Hole (GeänderteFelder)“ eine Liste von Feldern zurück, die durch das Script bearbeitet wurden.
- Um von der Art des Clients abhängige Scripts zu erstellen, verwenden Sie die Funktion „Hole(ProgrammVersion)“. Wenn der zurückgegebene Wert „Web Publishing Engine“ enthält, wissen Sie, dass der aktuelle Benutzer über Custom Web Publishing auf Ihre Datenbank zugreift. Weitere Informationen zu Funktionen finden Sie in der [FileMaker Pro Advanced Hilfe](#).
- Öffnen Sie jedes Script, das Webbenutzer ausführen könnten, und stellen Sie sicher, dass das Script richtig ausgeführt wird, wenn die Datenbank als Custom Web Publishing-Lösung bereitgestellt wird. Stellen Sie sicher, dass das Script nur Scriptschritte verwendet, die von Custom Web Publishing wie oben beschrieben unterstützt werden.

Scriptverhalten in Custom Web Publishing-Lösungen

Die Funktionsweise einiger Scriptschritte ist in Custom Web Publishing-Lösungen und in FileMaker Pro Advanced unterschiedlich. Informationen zur Kompatibilität finden Sie in der [FileMaker Pro Advanced Hilfe](#).

Scripts in Custom Web Publishing-Lösungen können keine Scripts in anderen FileMaker-Dateien ausführen, es sei denn, die Dateien werden auf derselben Installation von FileMaker Server bereitgestellt und dasselbe erweiterte Custom Web Publishing-Zugriffsrecht ist in den anderen Dateien aktiviert.

Script-Trigger in Custom Web Publishing-Lösungen

In FileMaker Pro Advanced können sowohl Scripts als auch Benutzeraktionen (wie zum Beispiel ein Benutzer, der in ein Feld klickt) Script-Trigger aktivieren. In Custom Web Publishing können jedoch nur Scripts Script-Trigger aktivieren.

Informationen zu Script-Trigger finden Sie in der [FileMaker Pro Advanced Hilfe](#).

Hinweis Bei Custom Web Publishing-Lösungen wird der Script-Trigger „BeiErstemFensterÖffnen“ nicht aktiviert. Der Script-Trigger „BeiLetztemFensterSchließen“ wird nur aktiviert, wenn ein Script das letzte virtuelle Fenster schließt. Sie können Scripts manuell mithilfe des XML-Abfrageparameter-Scripts oder der PHP-Methode `newPerformScriptCommand()` ausführen.

Kapitel 3

Custom Web Publishing mit XML

Erstellen dynamischer Webseiten mit der Web Publishing Engine

Die Web Publishing Engine bietet Custom Web Publishing für FileMaker Server mithilfe von XML-Datenveröffentlichung. Custom Web Publishing bietet mehrere Vorteile:

- **Anpassung:** Sie können festlegen, wie Webbenutzer mit FileMaker-Daten interagieren und wie die Daten in Webbrowsern angezeigt werden.
- **Datenaustausch:** Mithilfe von FileMaker XML können Sie FileMaker-Daten mit anderen Webseiten und Programmen austauschen.
- **Datenintegration:** Sie können FileMaker-Daten in andere Websites, mit anderer Middleware und mit eigenen Anwendungen integrieren. Sie können die Daten so darstellen, als würden sie zu einer anderen Website gehören, anstatt ein vollständiges FileMaker-Layout im Webbrowser anzuzeigen.
- **Sicherheit:** Der FileMaker Server-Administrator kann XML Web Publishing für alle Datenbanken, die der Server bereitstellt, aktivieren oder deaktivieren. Als Eigentümer der FileMaker-Datenbank können Sie den Benutzerzugriff auf XML Web Publishing für jede Datenbank steuern.
- **Steuern und Filtern von veröffentlichten Daten:** Sie können die Daten und die Art der Datenbankinformationen, die Sie veröffentlichen wollen, steuern und filtern und damit unbefugte Nutzung der Datenbank verhindern. Sie können auch Metadaten ausblenden wie z. B. Datenbank- und Feldnamen.
- **Basierend auf offenen Standards:** Ihnen stehen mehr Tools, Ressourcen und Fachpersonal für Custom Web Publishing-Lösungen zur Verfügung. Wenn Sie Standard-XML kennen, können Sie sofort mit der Entwicklung von Lösungen beginnen, nachdem Sie ein paar spezielle Details über Custom Web Publishing mit XML gelernt haben, z. B. die erforderliche URL-Syntax und Abfrage-Parameter.

Mit Custom Web Publishing mit XML können Sie Daten aus FileMaker-Datenbanken abrufen und die Daten einfach in anderen Ausgabeformaten verwenden. Mithilfe einer HTTP-Abfrage mit den geeigneten Abfrage-Befehlen und -Parametern können Sie FileMaker-Daten als ein XML-Dokument abrufen. Sie können die XML-Daten in anderen Anwendungen verwenden. Siehe „Zugriff auf XML-Daten über die Web Publishing Engine“ auf Seite 29.

Wichtige Funktionen in Custom Web Publishing mit XML

FileMaker Server Custom Web Publishing mit XML bietet mehrere wichtige neue Funktionen:

- Datenbanken werden auf FileMaker Server bereitgestellt und FileMaker Pro Advanced muss nicht ausgeführt werden.
- Sie können die serverseitige Verarbeitung von XML mit JavaScript nutzen.
- Wie in FileMaker Pro Advanced basiert der Zugriff auf Daten, Layouts und Felder auf den Benutzerkontoeinstellungen, die in den Zugriffsrechten der Datenbank definiert sind. Die Web Publishing Engine unterstützt auch mehrere andere Sicherheitsverbesserungen. Siehe „Schützen veröffentlichter Datenbanken“ auf Seite 14.
- Webbenutzer können komplexe Scripts mit mehreren Schritten ausführen. FileMaker unterstützt viele 70 Scriptschritte in Custom Web Publishing. Siehe „FileMaker-Scripts und Custom Web Publishing“ auf Seite 18.
- Sie können einen Parameterwert an ein FileMaker-Script übergeben. Siehe „Abfrage-Parameter –script.param (Parameter an Script übergeben)“ auf Seite 61, „Abfrage-Parameter –script.prefind.param (Parameter an Script vor Suchen übergeben)“ auf Seite 62 und „Abfrage-Parameter –script.presort.param (Parameter an Script vor Sortieren übergeben)“ auf Seite 62.
- Mit der `fmresultset`-XML-Grammatik können Sie nach Name auf Felder zugreifen und `relatedset`- (Ausschnitt) Daten manipulieren.
- Für den Zugriff auf Daten in einer Datenbank müssen Sie ein Layout angeben. Siehe Kapitel 5 „Gültige Namen in XML-Abfrage-Strings“.

Voraussetzungen für Web Publishing

Voraussetzungen für das Veröffentlichen einer Datenbank mit Custom Web Publishing

Um Datenbanken mithilfe von Custom Web Publishing mit XML zu veröffentlichen, benötigen Sie:

- einen FileMaker Server-Einsatz, der Folgendes umfasst:
 - einen Web-Server, entweder Microsoft IIS (Windows) oder Apache (macOS)
 - den FileMaker Datenbank-Server, aktiviert für Custom Web Publishing mit CLI
 - die Web Publishing Engine, installiert und konfiguriert
- eine oder mehrere FileMaker Pro-Advanced Datenbanken, die FileMaker Server bereitstellt
- IP-Adresse oder Domänenname des Hosts, auf dem der Web-Server läuft
- einen beliebigen Webbrowser und Zugriff auf den Web-Server, um Ihre Custom Web Publishing-Lösung zu entwickeln und zu testen

Weitere Informationen finden Sie im Handbuch [FileMaker Server Installation und Konfiguration](#).

Voraussetzungen für Webbenutzer zum Zugriff auf eine Custom Web Publishing-Lösung

Für den Zugriff auf eine Custom Web Publishing-Lösung mit XML benötigen Webbenutzer:

- einen Webbrowser
- Zugang zum Internet oder zu einem Intranet und zum Web-Server
- IP-Adresse oder Domänenname des Hosts, auf dem der Web-Server läuft

Wenn die Datenbank passwortgeschützt ist, müssen Webbenutzer auch einen Benutzernamen und ein Passwort für ein Datenbankkonto eingeben.

Verbindung zum Internet oder zu einem Intranet

Wenn Sie Datenbanken im Internet oder in einem Intranet veröffentlichen, muss auf dem Host-Computer FileMaker Server laufen und die freizugebenden Datenbanken müssen bereitgestellt und verfügbar sein. Zusätzlich:

- Veröffentlichen Sie Ihre Datenbank auf einem Computer, der über eine ständige Verbindung zum Internet bzw. Intranet verfügt. Sie können Datenbanken zwar ohne ständige Verbindung veröffentlichen, aber sie stehen Webbenutzern nur zur Verfügung, wenn Ihr Computer mit dem Internet bzw. Intranet verbunden ist.
- Der Host-Computer für den Web-Server, der Teil des FileMaker Server-Einsatzes ist, muss über eine eigene *statische* (permanente) IP-Adresse oder einen Domännennamen verfügen. Wenn Sie die Verbindung zum Internet über einen Internet-Anbieter (ISP) herstellen, könnte Ihre IP-Adresse *dynamisch zugewiesen* werden (d. h., sie ändert sich bei jeder Verbindung). Mit einer dynamischen IP-Adresse ist es für Webbenutzer schwieriger, Ihre Datenbanken zu finden. Wenn Sie sich über Ihre Zugangsart nicht sicher sind, wenden Sie sich an Ihren ISP oder Netzwerkadministrator.

Die nächsten Schritte

Hier einige Vorschläge, um mit der Entwicklung von Custom Web Publishing-Lösungen zu beginnen:

- Falls noch nicht geschehen, aktivieren Sie Custom Web Publishing mittels CLI. Weitere Informationen finden Sie in der [FileMaker Server Hilfe](#).
- Öffnen Sie in FileMaker Pro Advanced jede FileMaker-Datenbank, die Sie veröffentlichen wollen, und stellen Sie sicher, dass in der Datenbank die geeigneten erweiterten Zugriffsrechte für Custom Web Publishing aktiviert sind. Siehe „Aktivieren von Custom Web Publishing in einer Datenbank“ auf Seite 13.
- Wie Sie mithilfe von XML auf Daten in FileMaker-Datenbanken zugreifen, erfahren Sie in „Zugriff auf XML-Daten über die Web Publishing Engine“ auf Seite 29.

Kapitel 4

Zugriff auf XML-Daten mit der Web Publishing Engine

Sie können FileMaker-Daten in Extensible Markup Language- (XML) Format mithilfe der Web Publishing Engine beziehen und aktualisieren. Diese Sprache wird von zahlreichen Einzelpersonen, Organisationen und Unternehmen zur Übermittlung von Produktinformationen, Transaktionen, Inventardaten und anderen geschäftlichen Informationen genutzt.

Verwenden von Custom Web Publishing mit XML

Wenn Sie Standard-XML kennen, können Sie mit dem Einsatz der Web Publishing Engine beginnen, nachdem Sie ein paar spezielle Details über Custom Web Publishing mit XML wie z. B. die erforderliche URL-Syntax und Abfrage-Parameter gelernt haben.

Indem Sie HTTP-URL-Abfragen mit FileMaker-Abfrage-Befehlen und -Parametern verwenden, können Sie eine Datenbank abfragen, die von FileMaker Server bereitgestellt wird, und die resultierenden Daten in XML-Format herunterladen. Sie können eine Datenbank beispielsweise auf alle Datensätze in einem bestimmten Postleitzahlenbereich abfragen und die resultierenden XML-Daten wie gewünscht verwenden.

Weitere Informationen finden Sie in der [FileMaker Knowledge Base](#).

Hinweis Die von der Web Publishing Engine generierten XML-Daten sind wohlgeformt und richten sich nach der XML-1.0-Spezifikation. Informationen zu den Anforderungen des XML-Aufbaus finden Sie in der XML-Spezifikation unter www.w3.org.

Vergleich der Web Publishing Engine mit XML-Import und -Export

Die Web Publishing Engine und FileMaker Pro Advanced gestatten Ihnen beide, XML-Daten mit FileMaker-Datenbanken zu verwenden. Es gibt jedoch einige wesentliche Unterschiede zwischen den beiden Methoden:

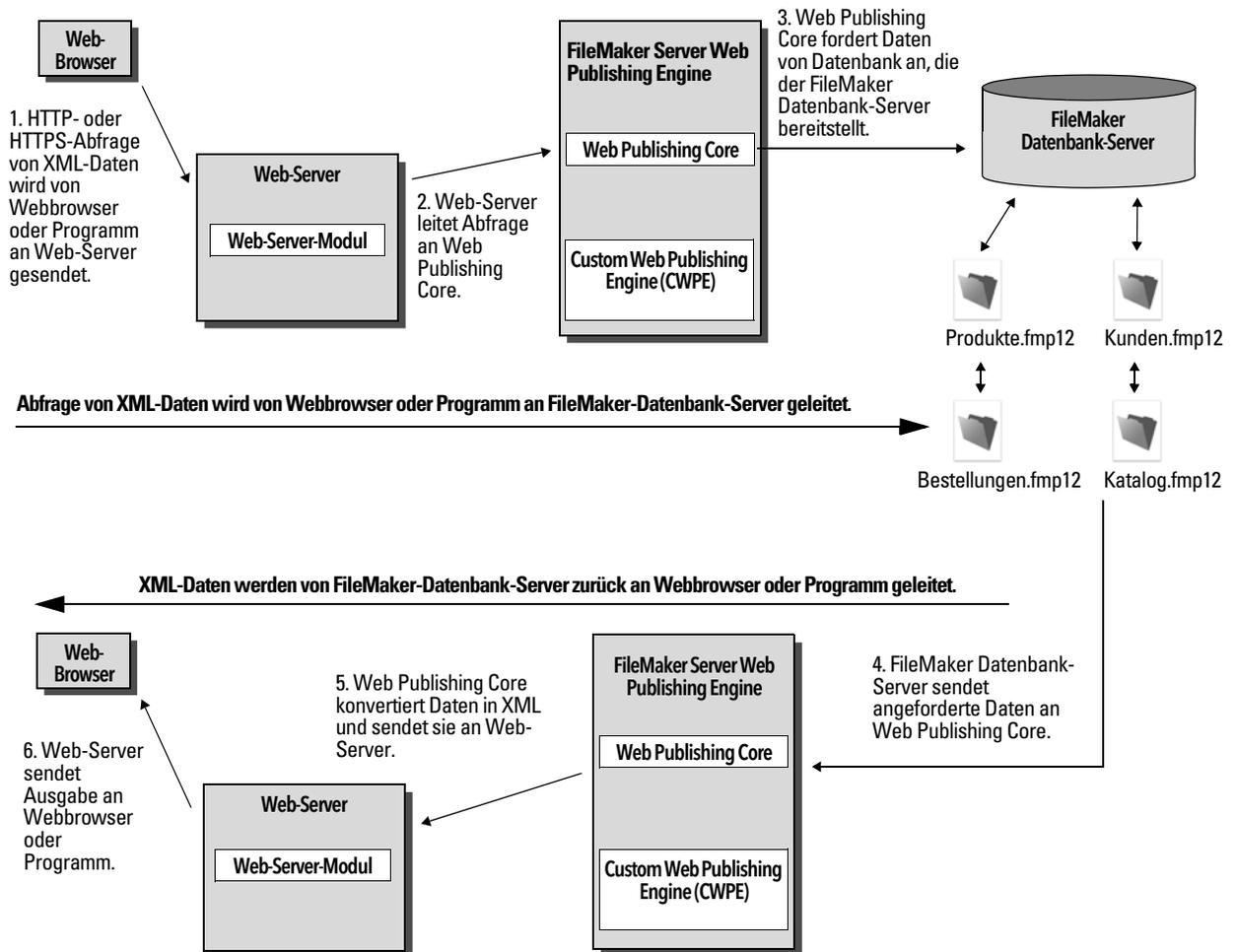
- Für den Zugriff auf XML-Daten unterstützt die Web Publishing Engine die `fmresultset`-, die `FMPXMLRESULT`- und die `FMPXMLLAYOUT`-Grammatik. Für den XML-Import verwendet FileMaker Pro Advanced die `FMPXMLRESULT`-Grammatik und für den Export die `FMPXMLRESULT`-Grammatik. Siehe „Zugriff auf XML-Daten über die Web Publishing Engine“ auf Seite 29.
- Um mit der Web Publishing Engine auf XML-Daten zuzugreifen, verwenden Sie einen Web Publishing Engine-Abfrage-String in einem URL. Für den Import und Export von XML mit FileMaker Pro Advanced verwenden Sie FileMaker Pro Advanced-Menübefehle oder Scripts.
- Die Web Publishing Engine ist serverbasiert und kann auf demselben oder einem anderen Host als FileMaker Server installiert werden. FileMaker Pro Advanced XML-Import und -Export ist desktopbasiert.
- Sie können dynamisch auf XML-Daten von FileMaker-Datenbanken zugreifen, indem Sie URL-Abfragen mit der Web Publishing Engine verwenden. Die XML-Exportfunktion von FileMaker Pro Advanced erzeugt eine zuvor angegebene XML-Datendatei.

- Das Arbeiten mit XML-Daten über die Web Publishing Engine ist ein interaktiver Vorgang. FileMaker Pro Advanced XML-Import und -Export erfolgt im Stapelbetrieb.
- Die Web Publishing Engine kann auf XML-Daten von einem FileMaker-Ausschnitt zugreifen, FileMaker Pro Advanced nicht.
- Die Web Publishing Engine kann auf Daten in einem Containerfeld zugreifen, FileMaker Pro Advanced nicht.
- Die Web Publishing Engine bietet Echtzeitzugriff auf FileMaker-Daten über HTTP oder HTTPS, FileMaker Pro Advanced nicht.

Hinweis Informationen über das Verwenden von FileMaker Pro Advanced für den Import und Export von Daten in XML-Format finden Sie in der [FileMaker Pro Advanced Hilfe](#).

Wie die Web Publishing Engine aus einer Abfrage XML-Daten generiert

Nachdem eine Abfrage von XML-Daten an den Web-Server gesendet wurde, fragt die Web Publishing Engine die FileMaker-Datenbank ab und gibt die Daten als XML-Dokument zurück.



Allgemeines Verfahren beim Zugriff auf XML-Daten von der Web Publishing Engine

Im Folgenden erhalten Sie einen Überblick über das Verfahren beim Verwenden der Web Publishing Engine für den Zugriff auf XML-Daten in einer FileMaker-Datenbank:

1. Stellen Sie mithilfe der CLI sicher, dass XML Publishing aktiviert ist. Weitere Informationen finden Sie in der [FileMaker Server Hilfe](#).
2. Öffnen Sie in FileMaker Pro Advanced jede FileMaker-Datenbank, die Sie veröffentlichen wollen, und stellen Sie sicher, dass in der Datenbank das erweiterte Zugriffsrecht fmxml für XML Custom Web Publishing aktiviert ist. Siehe „Aktivieren von Custom Web Publishing in einer Datenbank“ auf Seite 13.

Um auf XML-Daten in einem Ausschnitt zuzugreifen, stellen Sie die Ansicht für das Datenbanklayout auf **Als Formular anzeigen** oder **Als Liste anzeigen** ein. Wenn ein Benutzer oder Script die Ansicht des Datenbanklayouts zu **Als Tabelle anzeigen** ändert, kann nur auf den ersten Bezugsdatensatz (erste Zeile im Ausschnitt) als XML-Daten zugegriffen werden.

Die XML-Daten werden in einer Reihenfolge ausgegeben, die der Reihenfolge entspricht, in der die Feldobjekte dem Layout hinzugefügt wurden. Wenn die XML-Datenreihenfolge der Reihenfolge entsprechen soll, in der die Felder am Bildschirm angezeigt werden (von oben nach unten, von links nach rechts), dann wählen Sie alle Felder aus, gruppieren Sie sie und heben Sie dann die Gruppierung wieder auf. Dieser Vorgang setzt die Layoutreihenfolge wieder zurück auf die Bildschirmreihenfolge.

3. Bestätigen Sie in FileMaker Server Admin Console, dass für jede FileMaker-Datenbank, die Sie veröffentlichen, das erweiterte Zugriffsrecht fmxml aktiviert ist.
 - Klicken Sie in Admin Console auf die Seite **Datenbanken**.
 - Wählen Sie **Erweiterte Zugriffsrechte einblenden** aus der Menüauswahl neben **Alle Datenbanken**. Der Text **FMXML** wird für Datenbanken angezeigt, bei denen das erweiterte Zugriffsrecht fmxml aktiviert ist.
4. Senden Sie durch ein HTML-Formular, einen HREF-Link oder ein Script in Ihrem Programm oder Ihrer Webseite eine HTTP- oder HTTPS-Abfrage in Form eines URL mit Angabe der FileMaker XML-Grammatik, einen Abfrage-Befehl sowie einen oder mehrere FileMaker-Abfrage-Parameter an die Web Publishing Engine. Der URL kann auch in einem Browser eingegeben werden.

Informationen über die Angabe des URLs finden Sie im nächsten Abschnitt, „Erläuterung der URL-Syntax für XML-Daten und Containerobjekte“. Informationen über Abfrage-Befehle und -Parameter finden Sie unter „Abfragen von XML-Daten mithilfe von FileMaker-Abfrage-Strings“ auf Seite 39 und in Kapitel 5 „Gültige Namen in XML-Abfrage-Strings“.

5. Die Web Publishing Engine verwendet die im URL angegebene Grammatik, um XML-Daten mit den Ergebnissen Ihrer Abfrage zu erzeugen (z. B. eine Gruppe von Datensätzen aus der Datenbank), und gibt diese an Ihr Programm oder Ihren Webbrowser zurück.
6. Der Webbrowser, falls er über einen XML-Parser verfügt, zeigt die Daten an bzw. das Programm verwendet die Daten wie von Ihnen angegeben.

Erläuterung der URL-Syntax für XML-Daten und Containerobjekte

Dieser Abschnitt beschreibt die URL-Syntax beim Verwenden der Web Publishing Engine für den Zugriff auf XML-Daten und Containerobjekte in FileMaker-Datenbanken.

Erläuterung der URL-Syntax für XML-Daten

Die URL-Syntax beim Verwenden der Web Publishing Engine für den Zugriff auf XML-Daten in FileMaker-Datenbanken lautet:

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

Dabei gilt:

- `<scheme>` kann das HTTP- oder HTTPS-Protokoll sein.
- `<host>` ist die IP-Adresse oder der Domänenname des Hosts, auf dem der Web-Server installiert ist.
- `<port>` ist optional und gibt den Port an, den der Web-Server verwendet. Ohne Portangabe wird der Standard-Port des Protokolls (Port 80 für HTTP oder Port 443 für HTTPS) verwendet.
- `<xml_grammar>` ist der Name der FileMaker-XML-Grammatik. Mögliche Werte sind `fmresultset`, `FMPXMLRESULT` bzw. `FMPXMLLAYOUT`. Weitere Informationen hierzu finden Sie unter „Verwenden der `fmresultset`-Grammatik“ auf Seite 30 und „Verwenden anderer FileMaker XML-Grammatiken“ auf Seite 34.
- `<query string>` ist eine Kombination aus einem Abfrage-Befehl und einem oder mehreren Abfrage-Parametern für FileMaker XML Publishing. (Der Befehl `-dbnames` verlangt keine Parameter.) Weitere Informationen hierzu finden Sie unter „Abfragen von XML-Daten mithilfe von FileMaker-Abfrage-Strings“ auf Seite 39 und Kapitel 5 „Gültige Namen in XML-Abfrage-Strings“.

Hinweis Die URL-Syntax einschließlich der Namen von Abfrage-Befehl und -Parametern unterscheidet Groß- und Kleinschreibung, mit Ausnahme von Teilen des Abfrage-Strings. Der URL ist vorwiegend in Kleinbuchstaben mit Ausnahme der zwei Grammatiknamen in Großbuchstaben: `FMPXMLRESULT` und `FMPXMLLAYOUT`. Informationen über die Regeln für Groß-/Kleinschreibung im Abfrage-String finden Sie unter „Richtlinien für die Verwendung von Abfrage-Befehlen und -Parametern“ auf Seite 44.

Beispiele

```
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
```

Erläuterung der URL-Syntax für FileMaker-Containerobjekte in XML-Lösungen

In einem generierten XML-Dokument für eine XML-Lösung unterscheidet sich die Syntax für den Verweis auf ein Containerobjekt für Containerfelder, die das tatsächliche Objekt in der Datenbank speichern, und für Containerfelder, die einen Verweis auf das Objekt speichern.

Wenn ein Containerfeld das tatsächliche Objekt in der Datenbank speichert

Das `<data>`-Element des Containerfelds verwendet die folgende relative URL-Syntax, um auf das Objekt zu verweisen:

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

Dabei gilt: `<extension>` ist die Dateinamenerweiterung zur Identifizierung des Objekttyps, beispielsweise `.jpg`. Die Dateinamenerweiterung stellt den MIME-Typ ein, damit der Webbrowser die Containerdaten korrekt identifizieren kann. Informationen über `<query string>` finden Sie im vorherigen Abschnitt, „Erläuterung der URL-Syntax für XML-Daten“.

Beispiel:

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

Hinweis In der generierten XML für ein Containerfeld ist der Wert für den Abfrage-Parameter `-field` ein vollständig qualifizierter Feldname. Die Zahl in Klammern gibt die Wiederholungsnummer für das Containerfeld an und wird für Wiederhol- und für Einzelfelder generiert. Siehe „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 46.

Verwenden Sie zum Abrufen der Containerdaten aus der Datenbank die folgende Syntax:

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

Informationen über `<scheme>`, `<host>` oder `<port>` finden Sie im vorherigen Abschnitt, „Erläuterung der URL-Syntax für XML-Daten“.

Beispiel:

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

Wenn ein Containerfeld einen Dateiverweis anstelle eines tatsächlichen Objekts speichert

Das `<data>`-Element des Containerfelds enthält einen relativen Pfad, der auf das Objekt verweist.

Beispiel:

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

Hinweis Das Containerobjekt, auf das verwiesen wird, muss im Ordner „FileMaker Pro Advanced Web“ gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird, und dann in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software kopiert oder verschoben werden. Siehe „Veröffentlichen des Inhalts von Containerfeldern im Web“ auf Seite 16.

Wenn ein Containerfeld leer ist

Das `<data>`-Element des Containerfelds ist leer.

Erläuterung von URL-Textkodierung

Die URLs für den Zugriff auf XML-Daten und Containerfelder müssen in UTF-8 (Unicode Transformation 8 Bit)-Format kodiert werden. Weitere Informationen finden Sie unter „Erläuterung von UTF-8-kodierten Daten“ auf Seite 39.

Beispiel:

Um den Wert des „Info“-Felds zu *Matinée* zu ändern, könnten Sie folgenden URL verwenden:

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=members
&-lay=relationships&-recid=2&info= matin%C3%A9e&-edit
%C3%A9 ist die URL-kodierte UTF-8-Darstellung des Zeichens é.
```

Weitere Informationen finden Sie in der URL-Spezifikation unter www.w3.org.

Zugriff auf XML-Daten über die Web Publishing Engine

Um über die Web Publishing Engine auf XML-Daten zuzugreifen, verwenden Sie einen URL, der den Namen der zu verwendenden FileMaker-Grammatik, einen FileMaker-Abfrage-Befehl und einen oder mehrere FileMaker-Abfrage-Parameter angibt. Die Web Publishing Engine generiert XML-Daten aus Ihrer Datenbank, die nach einem der folgenden Typen von XML-Grammatiken formatiert werden:

- **fmresultset:** Dies ist die empfohlene Grammatik für die Web Publishing Engine für den Zugriff auf XML-Daten. Sie ist flexibel und für bequemen Feldzugriff nach Name und einfacherer Änderung von *relatedset*- (Ausschnitt-) Daten optimiert. Diese Grammatik ist auch direkter mit FileMaker-Terminologie verbunden und bietet z. B. globale Speicheroptionen und Identifikation von Statistik- und Formelfeldern. Für einfacheres Web Publishing wurde diese Grammatik knapper formuliert als die *FMPXMLRESULT*-Grammatik. Siehe „Verwenden der *fmresultset*-Grammatik“ auf Seite 30.
- **FMPXMLRESULT und FMPXMLLAYOUT:** Sie können auch die *FMPXMLRESULT*- und *FMPXMLLAYOUT*-Grammatiken mit der Web Publishing Engine für den Zugriff auf XML-Daten verwenden. Um ein einziges Stylesheet sowohl für XML-Export als auch für Custom Web Publishing einzusetzen, müssen Sie die *FMPXMLRESULT*-Grammatik verwenden. Für den Zugriff auf Wertelisten und Feldanzeigedaten in Layouts müssen Sie die *FMPXMLLAYOUT*-Grammatik verwenden. Siehe „Verwenden anderer FileMaker XML-Grammatiken“ auf Seite 34.

Abhängig von der Grammatik, die Sie in der URL-Abfrage angegeben haben, generiert die Web Publishing Engine mit einer der folgenden Grammatiken ein XML-Dokument. Jedes XML-Dokument enthält eine XML-Standard-Namespace-Deklaration für die Grammatik. Informationen finden Sie im nächsten Abschnitt, „Erläuterung von Namensräumen für FileMaker XML“. Verwenden Sie eine dieser Grammatiken in Ihrem Dokument oder auf Ihrer Webseite, um FileMaker-Daten im XML-Format anzuzeigen und zu bearbeiten.

Hinweis Von der Web Publishing Engine generierte XML-Daten werden im UTF-8-Format (Unicode Transformation Format 8) kodiert. Siehe „Erläuterung von UTF-8-kodierten Daten“ auf Seite 39.

Erläuterung von Namensräumen für FileMaker XML

Mithilfe eindeutiger XML-Namensräume können Sie XML-Tags der richtigen Anwendung zuordnen. Wenn Ihr XML-Dokument beispielsweise zwei <DATABASE>-Elemente enthält, eines für FileMaker-XML-Daten, das andere für Oracle-XML-Daten, kann anhand der Namensräume festgestellt werden, welches <DATABASE>-Element zu welchen XML-Daten gehört.

Die Web Publishing Engine generiert für jede Grammatik einen Standard-Namespace.

Grammatik	Generierter Standard-Namespace
fmresultset	xmlns="http://www.filemaker.com/xml/fmresultset"
FMPXMLRESULT	xmlns="http://www.filemaker.com/fmpxmlresult"
FMPXMLLAYOUT	xmlns="http://www.filemaker.com/fmpxmllayout"

Erläuterung der FileMaker-Datenbankfehlercodes

Die Web Publishing Engine gibt in den „error code“-Elementen am Beginn jedes XML-Dokuments einen Fehlercode für einen etwaigen Fehler zurück, der im zuletzt ausgeführten Abfrage-Befehl aufgetreten ist. Null (0) bedeutet keinen Fehler.

Grammatik	Verwendete Syntax
fmresultset	<error code="0"/error>
FMPXMLRESULT	ERRORCODE0/ERRORCODE
FMPXMLLAYOUT	ERRORCODE0/ERRORCODE

Das „error code“-Element im XML-Dokument gibt Fehler an, die die Datenbank und Abfrage-Strings betreffen. Siehe Anhang A „Fehlercodes für Custom Web Publishing“.

Abrufen der Document Type Definitions für die FileMaker-Grammatiken

Sie können die Document Type Definitions (DTDs) für die FileMaker-Grammatiken mithilfe einer HTTP-Abfrage abrufen.

Grammatik	HTTP-Abfrage
fmresultset	http://<host>[:<port>]/fmi/xml/fmresultset.dtd
FMPXMLRESULT	http://<host>[:<port>]/fmi/xml/FMPXMLRESULT.dtd
FMPXMLLAYOUT	http://<host>[:<port>]/fmi/xml/FMPXMLLAYOUT.dtd

Verwenden der fmresultset-Grammatik

Die XML-Elementnamen in dieser Grammatik verwenden FileMaker-Terminologie und die Speicherung von Feldern erfolgt getrennt von den Feldtypen. Die Grammatik umfasst auch die Fähigkeit, Statistik-, Formel- und Variablenfelder zu identifizieren.

Geben Sie zur Verwendung der fmresultset-Grammatik den folgenden Namen der fmresultset-Grammatik im URL an, der das XML-Dokument von der Web Publishing Engine anfordert:

```
fmresultset.xml
```

Beispiel:

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

Hinweis Geben Sie die `fmresultset`-Grammatik unbedingt in Kleinbuchstaben an.

Die Web Publishing Engine generiert ein XML-Dokument mithilfe der `fmresultset`-Grammatik. Im XML-Dokument verweist die Web Publishing Engine auf die Document Type Definition für die `fmresultset`-Grammatik in der Anweisung `<!DOCTYPE>` in der zweiten Zeile des Dokuments, unmittelbar nach der Anweisung `<?xml . . . ?>`. Die Anweisung `<!DOCTYPE>` gibt den URL für das Herunterladen der DTD für die `fmresultset`-Grammatik an.

Beschreibung der Elemente der `fmresultset`-Grammatik

Die `fmresultset`-Grammatik besteht hauptsächlich aus dem Element `<datasource>`, dem Element `<metadata>` und dem Element `<resultset>`.

`<datasource>`-Element

In der `fmresultset`-Grammatik enthält das `<datasource>`-Element die Attribute „table“, „layout“, „date-format“, „time-format“, „timestamp-format“, „total-count“ und „database“.

- Das Attribut „date-format“ des `<datasource>`-Elements gibt das Format von Datumsangaben im XML-Dokument an:

```
MM/dd/yyyy
```

Dabei gilt:

- `MM` ist der zweistellige Wert für den Monat (01 bis 12, wobei 01 für Januar und 12 für Dezember steht).
- `dd` ist der zweistellige Wert für den Tag des Monats (01 bis 31).
- `yyyy` ist der vierstellige Wert für das Jahr.
- Das Attribut „time-format“ des `<datasource>`-Elements gibt das Format von Zeitangaben im XML-Dokument an.

```
HH:mm:ss
```

Dabei gilt:

- `HH` ist der zweistellige Wert für Stunden (00 bis 23 im 24-Stunden-Format).
- `mm` ist der zweistellige Wert für Minuten (00 bis 59).
- `ss` ist der zweistellige Wert für Sekunden (00 bis 59).
- Das Attribut „timestamp-format“ des `<datasource>`-Elements gibt das Format von Datums- und Zeitangaben in einem Zeitstempel an.

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

`<metadata>`-Element

Das `<metadata>`-Element der `fmresultset`-Grammatik enthält ein oder mehrere `<field-definition>`- und `<relatedset-definition>`-Elemente, jeweils mit Attributen für eines der Felder in der Ergebnismenge.

Die `<field-definition>`-Attribute geben an:

- ob das Feld ein Feld mit automatischer Eingabe ist („yes“ oder „no“)
- ob das Feld ein Feld mit vierstelligem Jahr ist („yes“ oder „no“)
- ob das Feld ein Variablenfeld ist („yes“ oder „no“)
- die maximale Anzahl an Wiederholwerten (Attribut `max-repeat`)
- die maximale Anzahl an zulässigen Zeichen (Attribut `max-characters`)
- ob das Feld ein nicht leeres Feld ist („yes“ oder „no“)
- ob das Feld ein Feld nur mit numerischen Daten ist („yes“ oder „no“)
- `result` („text“, „number“, „date“, „time“, „timestamp“ oder „container“)
- ob das Feld ein `time-of-day`-Feld ist („yes“ oder „no“)
- `type` („normal“, „calculation“ oder „summary“)
- den Feldnamen (bei Bedarf vollständig qualifiziert)

Das `<relatedset-definition>`-Element repräsentiert einen Ausschnitt. Jedes Bezugsfeld in einem Ausschnitt wird durch das `<field-definition>`-Element innerhalb des `<relatedset-definition>`-Elements dargestellt. Wenn ein Ausschnitt mehrere Bezugsfelder enthält, werden die Felddefinitionen für die Bezugsfelder in einem einzigen `<relatedset-definition>`-Element gruppiert.

<resultset>-Element

Das `<resultset>`-Element enthält alle `<record>`-Elemente, die als Ergebnis einer Abfrage zurückgegeben wurden, sowie ein Attribut für die Gesamtzahl der gefundenen Datensätze. Jedes `<record>`-Element enthält die Felddaten für einen Datensatz in der Ergebnismenge – einschließlich der Attribute `mod-id` und `record-id` für den Datensatz – sowie das `<data>`-Element mit den Daten für ein Feld im Datensatz.

Jeder Datensatz in einem Ausschnitt wird durch ein `<record>`-Element innerhalb des `<relatedset>`-Elements dargestellt. Das Attribut „count“ des `<relatedset>`-Elements gibt die Anzahl der Datensätze in einem Ausschnitt an und das Attribut „table“ gibt die mit dem Ausschnitt verbundene Tabelle an.

XML-Daten in der fmresultset-Grammatik

Beispiel:

```
<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>
```

Verwenden anderer FileMaker XML-Grammatiken

Die anderen FileMaker XML-Grammatiken enthalten Informationen über Feldtypen, Wertelisten und Layouts. `FMPXMLRESULT` und `fmresultset` sind hinsichtlich der Funktionalität äquivalent. Für den Zugriff auf Wertelisten und Feldanzeigedaten in Layouts müssen Sie die `FMPXMLLAYOUT`-Grammatik verwenden. Die `FMPXMLRESULT`- und `FMPXMLLAYOUT`-Grammatiken sind kompakter für den Datenaustausch.

Geben Sie zur Verwendung der `FMPXMLRESULT`-Grammatik den folgenden Grammatiknamen im URL an, der das XML-Dokument von der Web Publishing Engine anfordert:

```
FMPXMLRESULT.xml
```

Beispiel:

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

Geben Sie zur Verwendung der `FMPXMLLAYOUT`-Grammatik den folgenden Grammatiknamen mit dem Abfrage-Befehl `-view` im URL an, der das XML-Dokument von der Web Publishing Engine anfordert:

```
FMPXMLLAYOUT.xml
```

Beispiel:

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

Hinweis Geben Sie die `FMPXMLRESULT`- und die `FMPXMLLAYOUT`-Grammatik unbedingt in Großbuchstaben an.

Im generierten XML-Dokument verweist die Web Publishing Engine auf die Document Type Definition für die Grammatik in der Anweisung `<!DOCTYPE>` in der zweiten Zeile des Dokuments, unmittelbar nach der Anweisung `<?xml . . . ?>`. Die Anweisung `<!DOCTYPE>` gibt den URL für das Herunterladen der DTD für die Grammatik an.

Beschreibung der Elemente in der `FMPXMLRESULT`-Grammatik

In der `FMPXMLRESULT`-Grammatik enthält das `<DATABASE>`-Element die Attribute `NAME`, `RECORDS`, `DATEFORMAT`, `LAYOUT` und `TIMEFORMAT`.

Das Attribut `DATEFORMAT` des `<DATABASE>`-Elements gibt das Datumsformat im XML-Dokument an. Das Attribut `TIMEFORMAT` des `<DATABASE>`-Elements gibt das Zeitformat im XML-Dokument an. Datums- und Zeitformate der Grammatiken `FMPXMLRESULT` und `fmresultset` sind gleich, Siehe „Beschreibung der Elemente der `fmresultset`-Grammatik“ auf Seite 31.

Das Element `<METADATA>` der `FMPXMLRESULT`-Grammatik enthält ein oder mehrere `<FIELD>`-Elemente. Jedes davon enthält Daten für ein Feld bzw. eine Spalte im Ergebnis – z. B. den Feldnamen, so wie er in der Datenbank definiert ist, den Feldtyp, die Ja/Nein-Zulässigkeit für leere Felder (Attribut `EMPTYOK`) und die maximale Anzahl von Wiederholwerten (Attribut `MAXREPEAT`). Gültige Werte für Feldtypen sind `TEXT`, `NUMBER`, `DATE`, `TIME`, `TIMESTAMP` und `CONTAINER`.

Das `<RESULTSET>`-Element enthält alle `<ROW>`-Elemente, die als Ergebnis einer Abfrage zurückgegeben wurden, sowie ein Attribut für die Gesamtzahl der gefundenen Datensätze. Jedes `<ROW>`-Element enthält die Feld-/Spalten-Daten für jede Zeile in der Ergebnismenge. Diese Daten umfassen RECORDID und MODID für die Zeile (siehe „Abfrage-Parameter –modid (ÄnderungsID)“ auf Seite 57) und das `<COL>`-Element. Das `<COL>`-Element enthält die Daten für ein Feld/eine Spalte in der Zeile, wobei mehrere `<DATA>`-Elemente einen der Werte in einem Wiederhol- oder Ausschnittfeld repräsentieren.

XML-Daten in der FMPXMLRESULT-Grammatik

Beispiel:

```
FMPXMLRESULT xmlns="http://www.filemaker.com/fmpxmlresult"
  ERRORCODE0/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>
```

Die Reihenfolge der `<COL>`-Elemente entspricht der Reihenfolge der `<FIELD>`-Elemente im `<METADATA>`-Element – wo zum Beispiel die Felder „Title“ und „Artist“ im `<METADATA>`-Element aufgelistet werden, werden „Village Market“ und dann „Camille Pissarro“ in der gleichen Reihenfolge in den Elementen `<RESULTSET>` und `<ROW>` aufgelistet.

Beschreibung der Elemente der Grammatik FMPXMLLAYOUT

In der FMPXMLLAYOUT-Grammatik enthält das LAYOUT-Element den Namen des Layouts, den Namen der Datenbank sowie <FIELD>-Elemente für jedes Feld im entsprechenden Layout der Datenbank. Jedes <FIELD>-Element beschreibt den Feldtyp und enthält das Attribut VALUELIST für jede Werteliste des Felds.

Das Element <VALUELISTS> enthält ein oder mehrere <VALUELIST>-Elemente für jede Werteliste im Layout – jedes davon enthält den Namen der Werteliste und ein <VALUE>-Element für jeden Wert in der Liste.

Abhängig von den im Dialogfeld **Felder für Werteliste angeben** in der FileMaker-Datenbank gewählten Optionen enthält das Element <VALUE> ein DISPLAY-Attribut, das den Wert nur im ersten Feld, im zweiten Feld oder in beiden Feldern einer Werteliste enthält. Nehmen Sie zum Beispiel an, dass das erste Feld in einer Werteliste die ID-Nummer des Kunststils (wie z. B. „100“) und das zweite Feld den zugehörigen Namen des Kunststils (wie z. B. „Impressionismus“) enthält. Hier finden Sie eine Zusammenfassung des Inhalts des Attributs DISPLAY, wenn die verschiedenen Kombinationen der Optionen im Dialogfeld **Felder für Werteliste angeben** ausgewählt sind:

- Wenn **Auch Werte aus zweitem Feld anzeigen** nicht ausgewählt ist, enthält das Attribut DISPLAY nur den Wert im ersten Feld einer Werteliste.

Beispiel:

Das Attribut DISPLAY enthält nur die ID-Nummer des Kunststils:

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

- Wenn sowohl **Auch Werte aus zweitem Feld anzeigen** als auch **Werte nur aus dem zweiten Feld anzeigen** ausgewählt ist, enthält das DISPLAY-Attribut nur den Wert im zweiten Feld.

Beispiel:

Das Attribut DISPLAY enthält nur den Namen des Kunststils:

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

- Wenn **Auch Werte aus zweitem Feld anzeigen** ausgewählt ist und **Werte nur aus dem zweiten Feld anzeigen** nicht ausgewählt ist, enthält das DISPLAY-Attribut die Werte in beiden Feldern einer Werteliste.

Beispiel:

Das Attribut DISPLAY enthält sowohl die ID-Nummer als auch den Namen des Kunststils:

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

Für Datums-, Zeit- und Zeitstempelfelder werden Daten für Wertelisten über das „fm“-Format für diesen Feldtyp formatiert. Die „fm“-Formate sind MM/DD/YYYY für Datum, hh:mm:ss für Zeit und MM/DD/YYYY hh:mm:ss für Zeitstempel. Wenn zum Beispiel eine Werteliste „Geburtstage“ für ein Einblendmenü für ein Feld „Geburtstag“ eines Layouts verwendet wird und das Feld „Geburtstag“ ein Datumsfeld ist, werden alle für diese Werteliste ausgegebenen Werte im Datumsformat „fm“ ausgegeben.

Hinweis Wenn zwei Felder mit unterschiedlichem Feldtyp in einem Layout die gleiche Werteliste verwenden, bestimmt das erste Feld das Format der Daten der Werteliste.

XML-Daten in der FMPXMLLAYOUT-Grammatik

Beispiel:

```
<FMPXMLLAYOUT xmlns="http://www.filemaker.com/fmpxmllayout">
  ERRORCODE0/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>
```

Erläuterung von UTF-8-kodierten Daten

Alle von der Web Publishing Engine generierten XML-Daten werden im UTF-8-Format (Unicode Transformation Format 8) kodiert. Dieses Format komprimiert ASCII-Zeichen im Unicode-Standardformat von 16 Bit auf 8 Bit. Für Unicode und UTF-8 sind XML-Parser erforderlich.

UTF-8-Kodierung unterstützt die direkte Darstellung der Werte 0-127 für den ASCII-Standardzeichensatz in der englischen Sprache und bietet Multibyte-Kodierungen für Unicode-Zeichen mit höheren Werten.

Hinweis Stellen Sie sicher, dass Sie einen Webbrowser oder ein Textverarbeitungsprogramm verwenden, das UTF-8-Dateien unterstützt.

Das Kodierungsformat UTF-8 hat die folgenden Eigenschaften:

- Alle ASCII-Zeichen sind aus einem Byte bestehende UTF-8-Zeichen. Eine gültige ASCII-Zeichenkette ist eine gültige UTF-8-Zeichenkette.
- Jedes Nicht-ASCII-Zeichen (beliebiges Zeichen mit gesetztem höherwertigem Bit) ist Teil eines Multibyte-Zeichens.
- Das erste Byte eines UTF-8-Zeichens gibt die Anzahl der zusätzlichen Bytes im Zeichen an.
- Das erste Byte eines Multibyte-Zeichens lässt sich leicht vom darauf folgenden Byte unterscheiden, wodurch sich der Beginn eines Zeichens von einer beliebigen Position in einem Datenstrom einfach ermitteln lässt.
- Eine Konvertierung zwischen UTF-8 und Unicode ist einfach vorzunehmen.
- Die UTF-8-Kodierung ist relativ kompakt. Bei Text, der einen hohen Prozentsatz an ASCII-Zeichen enthält, ist sie kompakter als Unicode. Im schlimmsten Fall ist ein UTF-8-String nur 50 % größer als der entsprechende Unicode-String.

Abfragen von XML-Daten mithilfe von FileMaker-Abfrage-Strings

Um XML-Daten aus einer FileMaker-Datenbank abzufragen, verwenden Sie die FileMaker-Abfrage-Befehle und -Parameter in einem Abfrage-String.

Beispiel:

Sie können den Abfrage-Befehl `-findall` im folgenden Abfrage-String in einem URL verwenden, um eine Liste aller Produkte aus einer FileMaker-Datenbank mit dem Namen „products“ anzufordern:

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

Ein Abfrage-String darf nur einen Abfrage-Befehl, z. B. `-new`, enthalten. Die meisten Abfrage-Befehle verlangen auch mehrere entsprechende Abfrage-Parameter im Abfrage-String. Beispielsweise verlangen alle Abfrage-Befehle außer `-dbnames` den Parameter `-db`, der die abzufragende Datenbank angibt.

Sie können Abfrage-Befehle und -Parameter auch in einem URL verwenden.

Dieser Abschnitt bietet eine Zusammenfassung der FileMaker Abfrage-Befehle und -Parameter. Weitere Informationen über ihre Verwendung in einem Abfrage-String finden Sie unter „Gültige Namen in XML-Abfrage-Strings“ auf Seite 43.

Verwenden Sie diesen Abfrage-Befehlsnamen	Zur Ausführung dieses Befehls
-dbnames	Namen aller bereitgestellten und im Web gemeinsam genutzten Datenbanken abrufen.
-delete	Datensatz löschen.
-dup	Datensatz duplizieren.
-edit	Datensatz bearbeiten.
-find	Einen Datensatz oder eine Gruppe von Datensätzen suchen.
-findall	Nach allen Datensätzen suchen.
-findany	Zufällig ermittelten Datensatz anzeigen.
-findquery	Komplexe oder zusammengesetzte Suchabfrage durchführen.
-layoutnames	Namen aller verfügbaren Layouts für eine bereitgestellte und im Web gemeinsam genutzte Datenbank abrufen.
-new	Neuen Datensatz hinzufügen.
-scriptnames	Namen aller verfügbaren Scripts für eine bereitgestellte und im Web gemeinsam genutzte Datenbank abrufen.
-view	Layout-Informationen aus einer Datenbank abrufen, wenn die <code>FMPXMLLAYOUT-</code> Grammatik angegeben ist. Ruft den Abschnitt <code><metadata></code> des XML-Dokuments und eine leere Datensatzmenge ab, wenn die <code>fmresultset-</code> oder <code>FMPXMLRESULT-</code> Grammatik angegeben ist.

Verwenden Sie diese Abfrage-Parameternamen	Mit diesen Abfrage-Befehlen
-db (Datenbankname)	Erforderlich für alle Abfrage-Befehle mit Ausnahme von <code>-dbnames</code>
-delete.related	Optional bei <code>-edit</code>
-field	Erforderlich zur Angabe eines Felds in einem URL für Containerabfragen. Siehe „Erläuterung der URL-Syntax für FileMaker-Containerobjekte in XML-Lösungen“ auf Seite 28.
fieldname	Mindestens ein Feldname ist für <code>-edit</code> erforderlich. Optional bei <code>-find</code> . Siehe „Abfrage-Parameter „fieldname“ (Nicht-Containerfeldname)“ auf Seite 53.
fieldname.op (Operator)	Optional bei <code>-find</code>
-lay (Layoutname)	Erforderlich für alle Abfrage-Befehle mit Ausnahme von <code>-dbnames</code> , <code>-layoutnames</code> und <code>-scriptnames</code> .
-lay.response (Layout für XML-Antwort wechseln)	Optional für alle Abfrage-Befehle mit Ausnahme von <code>-dbnames</code> , <code>-layoutnames</code> und <code>-scriptnames</code> .
-lop (logischer Operator)	Optional bei <code>-find</code>
-max (maximale Datensätze)	Optional bei <code>-find</code> , <code>-findall</code> und <code>-findquery</code>
-modid (ÄnderungsID)	Optional bei <code>-edit</code>
-query	Erforderlich bei zusammengesetzten <code>-findquery</code> -Suchabfragen
-recid (DatensatzID)	Erforderlich bei <code>-edit</code> , <code>-delete</code> , <code>-dup</code> . Optional bei <code>-find</code>

Verwenden Sie diese Abfrage-Parameternamen	Mit diesen Abfrage-Befehlen
<code>-relatedsets.filter</code>	Optional bei <code>-find</code> , <code>-findall</code> , <code>-findany</code> , <code>-edit</code> , <code>-new</code> , <code>-dup</code> und <code>-findquery</code>
<code>-relatedsets.max</code>	Optional bei <code>-find</code> , <code>-edit</code> , <code>-new</code> , <code>-dup</code> und <code>-findquery</code>
<code>-script</code> (Script ausführen)	Optional bei <code>-find</code> , <code>-findall</code> , <code>-findany</code> , <code>-new</code> , <code>-edit</code> , <code>-delete</code> , <code>-dup</code> , <code>-view</code> und <code>-findquery</code>
<code>-script.param</code> (einen Parameterwert an das durch <code>-script</code> angegebene Script übergeben)	Optional bei <code>-script</code> und <code>-findquery</code>
<code>-script.prefind</code> (Script vor <code>-find</code> , <code>-findany</code> und <code>-findall</code> ausführen)	Optional bei <code>-find</code> , <code>-findany</code> , <code>-findall</code> und <code>-findquery</code>
<code>-script.prefind.param</code> (einen Parameterwert an das durch <code>-script.prefind</code> angegebene Script übergeben)	Optional bei <code>-script.prefind</code> und <code>-findquery</code>
<code>-script.presort</code> (Script vor Sortierung ausführen)	Optional bei <code>-find</code> , <code>-findall</code> und <code>-findquery</code>
<code>-script.presort.param</code> (einen Parameterwert an das durch <code>-script.presort</code> angegebene Script übergeben)	Optional bei <code>-script.presort</code> und <code>-findquery</code>
<code>-skip</code> (Datensätze überspringen)	Optional bei <code>-find</code> , <code>-findall</code> und <code>-findquery</code>
<code>-sortfield.[1-9]</code> (Sortierfeld)	Optional bei <code>-find</code> , <code>-findall</code> und <code>-findquery</code>
<code>-sortorder.[1-9]</code> (Sortierfolge)	Optional bei <code>-find</code> , <code>-findall</code>

Wechseln des Layouts für eine XML-Antwort

Der Abfrage-Parameter `-lay` gibt das Layout an, das Sie beim Abfragen von XML-Daten verwenden wollen. Häufig eignet sich dasselbe Layout für die Verarbeitung der Daten, die von der Abfrage geliefert werden. In einigen Fällen sollten Sie die Daten in einem Layout durchsuchen, das Felder enthält, die aus Sicherheitsgründen in keinem Layout vorhanden sind, mit dem die Ergebnisse angezeigt werden. (Um Daten in einem Feld zu suchen, muss das Feld auf das Layout gesetzt werden, das Sie in der XML-Abfrage angeben.)

Um für die Anzeige der XML-Antwort ein anderes Layout als für die Verarbeitung der XML-Abfrage zu verwenden, können Sie den optionalen Abfrage-Parameter `-lay.response` angeben.

Beispiel:

Die folgende Abfrage sucht Werte größer als 100.000 im Feld „Salary“ im Layout „Budget“. Die gefundenen Daten werden im Layout „ExecList“ angezeigt, das das Feld „Salary“ nicht enthält.

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

Erläuterung der Verarbeitung einer XML-Abfrage

Es gibt mehrere Abfrage-Parameter, die die Verarbeitung einer XML-Abfrage und die Generierung eines XML-Dokuments beeinflussen.

FileMaker Server und die Web Publishing Engine verarbeiten eine XML-Abfrage in dieser Reihenfolge:

1. Verarbeitung des `-lay` -Abfrage-Parameters
2. Festlegen der globalen Feldwerte, die in der Abfrage angegeben sind (der Teil `.global=` des URLs)
3. Verarbeitung des Abfrage-Parameters `-script.prefind`, falls angegeben
4. Verarbeitung der Abfrage-Befehle wie z. B. `-find` oder `-new`
5. Verarbeitung des Abfrage-Parameters `-script.presort`, falls angegeben
6. Sortieren der Ergebnisdaten, falls eine Sortierung angegeben wurde
7. Verarbeitung des Abfrage-Parameters `-script`, falls angegeben
8. Verarbeitung des Abfrage-Parameters `-lay.response`, um das Layout zu wechseln, falls angegeben
9. Generieren des XML-Dokuments.

Wenn einer der obigen Schritte einen Fehlercode erzeugt, stoppt die Abfrageverarbeitung und etwaige nachfolgende Schritte werden nicht ausgeführt. Jedoch bleiben alle vorangehenden Schritte in der Abfrage ausgeführt.

Betrachten Sie z. B. eine Abfrage, die den aktuellen Datensatz löscht, die Datensätze sortiert und dann ein Script ausführt. Wenn der Parameter `-sortfield` ein nicht vorhandenes Feld angibt, löscht die Abfrage den aktuellen Datensatz und gibt dann Fehlercode 102 („Feld fehlt“) zurück, führt aber das Script nicht aus.

Fehlerbehebung beim XML-Dokumentzugriff

Wenn beim Zugriff auf XML-Dokumente mit der Web Publishing Engine Probleme auftreten, prüfen Sie Folgendes:

- Die erweiterten Zugriffsrechte in der Datenbank müssen auf „XML Custom Web Publishing“ eingestellt und einem Benutzerkonto zugewiesen sein. Siehe „Aktivieren von Custom Web Publishing in einer Datenbank“ auf Seite 13.
- Die Datenbank wird auf der Datenbank-Server-Komponente des FileMaker Server-Einsatzes bereitgestellt und von FileMaker Server geöffnet. Weitere Informationen finden Sie in der [FileMaker Server Hilfe](#).
- Prüfen Sie, ob der verwendete Datenbankkontoname und ggf. Ihr Passwort korrekt sind.
- Die Web-Server-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- Die Web Publishing Engine-Komponente des FileMaker Server-Einsatzes wird ausgeführt.
- XML Publishing ist über CLI aktiviert. Weitere Informationen finden Sie in der [FileMaker Server Hilfe](#).

Kapitel 5

Gültige Namen in XML-Abfrage-Strings

Dieser Anhang beschreibt die gültigen Namen für Abfrage-Befehle und deren Parameter, die Sie in einem XML-Abfrage-String verwenden können, wenn Sie über die Web Publishing Engine auf FileMaker-Daten zugreifen.

Erläuterung von Abfrage-Befehlen und -Parametern

Die folgende Liste enthält sämtliche Namen für Abfrage-Befehle und Abfrage-Parameter:

Abfrage-Befehlsnamen

-dbnames (siehe Seite 49)
-delete (siehe Seite 49)
-dup (siehe Seite 49)
-edit (siehe Seite 49)
-find, -findall, -findany (siehe Seite 50)
-findquery (siehe Seite 50)
-layoutnames (siehe Seite 51)
-new (siehe Seite 51)
-scriptnames (siehe Seite 52)
-view (siehe Seite 52)

Abfrage-Parameternamen

-db (siehe Seite 52)
-field (siehe Seite 53)
fieldname (siehe Seite 53)
fieldname.op (siehe Seite 54)
-lay (siehe Seite 55)
-lay.response (siehe Seite 55)
-lop (siehe Seite 56)
-max (siehe Seite 56)
-modid (siehe Seite 57)
-query (siehe Seite 57)
-recid (siehe Seite 58)
-relatedsets.filter (siehe Seite 59)
-relatedsets.max (siehe Seite 60)
-script (siehe Seite 60)
-script.param (siehe Seite 61)
-script.prefind (siehe Seite 61)
-script.prefind.param (siehe Seite 62)
-script.presort (siehe Seite 62)
-script.presort.param (siehe Seite 62)
-skip (siehe Seite 63)
-sortfield.[1-9] (siehe Seite 63)
-sortorder.[1-9] (siehe Seite 64)

Wichtig Der Parameter `-lay` für die Angabe eines Datenbanklayouts ist bei allen Abfrage-Befehlen erforderlich mit Ausnahme von `-dbnames`, `-layoutnames` und `-scriptnames`.

Richtlinien für die Verwendung von Abfrage-Befehlen und -Parametern

Beachten Sie folgende Richtlinien, wenn Sie Abfrage-Befehle und -Parameter in einem Abfrage-String verwenden:

- Ein Abfrage-String darf nur einen einzigen Abfrage-Befehl enthalten. Ein Abfrage-String kann z. B. `-new` enthalten, um einen neuen Datensatz hinzuzufügen, aber derselbe Abfrage-String kann nicht gleichzeitig `-new` und `-edit` enthalten.
- Die meisten Abfrage-Befehle verlangen mehrere entsprechende Abfrage-Parameter im Abfrage-String. Beispielsweise verlangen alle Abfrage-Befehle außer `-dbnames` den Parameter `-db`, der die abzufragende Datenbank angibt. Siehe dazu die Tabelle der erforderlichen Parameter in „Abfragen von XML-Daten mithilfe von FileMaker-Abfrage-Strings“ auf Seite 39.
- Geben Sie für Abfrage-Parameter und Feldnamen den jeweils gewünschten Wert an, z. B. `-db=employees`. Geben Sie für Abfrage-Befehle kein `=`-Zeichen oder einen Wert nach dem Befehlsnamen wie `-findall` an.
- Die Web Publishing Engine konvertiert alle reservierten Wörter in Kleinbuchstaben, hierzu zählen Abfrage-Befehle, Abfrage-Parameter und Befehlswerte, bei denen spezifische Werte erwartet werden (zum Beispiel: `-lop=and`, `-lop=or`, `-sortorder=ascend`, `-sortorder=descend`, `-max=all`).
- Datenbanknamen, Layoutnamen und Feldnamen, die in Abfrage-Strings verwendet werden, unterscheiden keine Groß- und Kleinschreibung, z. B. gibt `-lay=meinlayout` den Layoutnamen `MeinLayout` an.
- Es empfiehlt sich nicht, Punkte oder Klammern in Feldnamen zu verwenden. In bestimmten Fällen funktionieren Feldnamen mit Punkten, aber Feldnamen mit den folgenden Ausnahmen können nie verwendet werden:
 - Dem Punkt darf keine Zahl folgen. Beispielsweise ist `meinfeld.9` ein ungültiger Dateiname.
 - Dem Punkt darf nicht die Textzeichenfolge `op` (die beiden Buchstaben „o“ und „p“) folgen. Beispielsweise ist `meinfeld.op` ein ungültiger Feldname.
 - Dem Punkt darf nicht die Textzeichenfolge `global` (das Wort „global“) folgen. Beispielsweise ist `meinfeld.global` ein ungültiger Feldname.Auf Feldnamen, die eine dieser Ausnahmen enthalten, kann nicht über XML mithilfe einer HTTP-Abfrage zugegriffen werden. Diese Konstrukte sind für DatensatzIDs, wie in Abschnitt „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 46 beschrieben, reserviert.
- Für den Befehl `-find` wird die Groß-/Kleinschreibung eines Feldwerts nicht berücksichtigt. Sie können beispielsweise `Field1=Blue` oder `Field1=blue` verwenden. Für die Befehle `-new` und `-edit` wird die verwendete Schreibung für einen Feldwert beibehalten und exakt so in der Datenbank gespeichert wie im Abfrage-String angegeben. Beispiel: `LastName=Doe`.

Abfrage-Befehls-Parsing

Die Web Publishing Engine analysiert Abfrage-Befehle in der folgenden Reihenfolge und stoppt die Analyse von XML-Abfragen mit dem ersten Fehler. Wenn ein Fehlercode zurückgegeben wird, entspricht der Fehlercode dem ersten Fehler, der erkannt wird.

1. Hat die Abfrage einen Befehl und ist der Abfrage-Befehl gültig?

Es ist ein Fehler, wenn der Abfrage der Befehl fehlt oder sie einen unbekanntem Befehl verwendet.

Beispiel:

```
-database
```

2. Hat die Abfrage zwei Befehle?

Beispiel:

```
-find&-edit
```

3. Hat die Abfrage einen ungültigen Wert für einen Befehl oder Parameter?

Beispiel:

```
-lop=amd
```

4. Fehlt der Abfrage der erforderliche Datenbanknamen-Parameter (-db)?

5. Fehlt der Abfrage der erforderliche Layoutbanknamen-Parameter (-lay)?

6. Hat die Abfrage eine ungültige Sortierung?

7. Hat die Abfrage ungültige Feldparameter?

Hinweis Wenn eine Abfrage gültige, aber irrelevante Informationen enthält, wird die Abfrage ohne Fehler verarbeitet. Wenn Sie z. B. den Parameter `-lop` bei einem `-delete`-Befehl angeben, wird der Parameter `-lop` ignoriert, da er nicht dazu führt, dass die Abfrage ungültig oder zweideutig wird.

Informationen zu spezifischen zurückgegebenen Fehlercodes finden Sie unter Anhang A „Fehlercodes für Custom Web Publishing“.

Erläuterung der Syntax für einen voll qualifizierten Feldnamen

Ein vollständig qualifizierter Feldname identifiziert eine exakte Instanz eines Felds. Da Felder mit gleichen Namen auf unterschiedlichen Tabellen basieren können, müssen Sie vollständig qualifizierte Namen verwenden, um Fehler zu vermeiden.

Syntax für einen vollständig qualifizierten Feldnamen:

```
tabellen-name::feld-name(wiederholung-nummer).datensatz-id
```

Dabei gilt:

- `tabellen-name` ist der Name der Tabelle, die das Feld enthält. Der Tabellenname ist nur erforderlich, wenn sich das Feld nicht in der zugrunde liegenden Tabelle des im Abfrage-String angegebenen Layouts befindet.
- `feld-name(wiederholung-nummer)` ist der spezifische Wert in einem Wiederholfeld und nur für Wiederholfelder erforderlich. Die Wiederholungsnummer wird ab der Ziffer 1 gezählt. Beispielsweise gibt `feld-name(2)` den zweiten Wert in einem Wiederholfeld an. Wenn Sie keine Wiederholungsnummer für ein Wiederholfeld angeben, wird der erste Wert des Wiederholfelds verwendet. Die Wiederholungsnummer ist für die Abfrage-Befehle `-new` und `-edit` erforderlich, wenn sie Wiederholfelder verwenden, jedoch nicht für den Befehl `-find`.
- `datensatz-id` gibt die DatensatzID an. Diese ist nur erforderlich, wenn Sie einen Abfrage-String verwenden, um Datensätze in einem Ausschnitt hinzuzufügen oder zu bearbeiten. Weitere Informationen finden Sie in den Abschnitten „Hinzufügen von Datensätzen in einem Ausschnitt“ und „Bearbeiten von Datensätzen in einem Ausschnitt“. Die `datensatz-id` ist für die Abfrage-Befehle `-new` und `-edit` erforderlich, wenn sie Ausschnittfelder verwenden, jedoch nicht für den Befehl `-find`.

Hinweis Damit sie verfügbar sind, müssen Felder auf das Layout gesetzt werden, das Sie im Abfrage-String angeben.

Verwenden von Abfrage-Befehlen für Ausschnittfelder

Die folgenden Abschnitte beschreiben, wie Abfrage-Befehle für Ausschnittfelder funktionieren.

Hinzufügen von Datensätzen in einem Ausschnitt

Um einem Ausschnitt gleichzeitig mit dem Hinzufügen eines Ausgangsdatensatzes einen neuen Datensatz hinzuzufügen, verwenden Sie den Abfrage-Befehl `-new` und führen den folgenden Abfrage-String für die Abfrage aus:

- Verwenden Sie den vollständig qualifizierten Feldnamen für das Bezugsfeld im Ausschnitt.
- Geben Sie 0 als DatensatzID nach dem Namen des Bezugsfelds im Ausschnitt an.
- Geben Sie mindestens eines der Felder für den Ausgangsdatensatz an, bevor Sie das Bezugsfeld im Ausschnitt angeben.
- Geben Sie die Daten für das Abgleichsfeld (Schlüselfeld) im Ausgangsdatensatz an.

Beispiel:

Der folgende URL fügt einen neuen Employee-Ausgangsdatensatz für John Doe und gleichzeitig einen neuen Bezugsdatensatz für Jane im Ausschnitt hinzu. Der Name der Bezugstabelle ist „Dependents“ und der Name des Bezugsfelds im Ausschnitt lautet „Names“. Das Abgleichsfeld „ID“ speichert eine Angestelltenkennung.

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

Hinweis Sie können pro Abfrage nur einen Bezugsdatensatz hinzufügen.

Bearbeiten von Datensätzen in einem Ausschnitt

Zur Bearbeitung eines oder mehrerer Datensätze in einem Ausschnitt verwenden Sie den Befehl `-edit` und eine DatensatzID, um den Ausgangsdatensatz anzugeben, der die zu bearbeitenden Ausschnitt-Datensätze enthält. Geben Sie den entsprechenden zu bearbeitenden Ausschnitt-Datensatz anhand seiner DatensatzID in einem vollständig qualifizierten Feldnamen an. Sie können eine DatensatzID dem DatensatzID-Attribut des `<record>`-Elements im `<relatedset>`-Element in den XML-Daten entnehmen. Siehe „Verwenden der fmresultset-Grammatik“ auf Seite 30.

Beispiele

Der folgende URL bearbeitet einen Datensatz in einem Ausschnitt, wobei der Ausgangsdatensatz die DatensatzID 1001 hat. Der Name der Bezugstabelle ist „Dependents“, der Name des Bezugsfelds im Ausschnitt lautet „Names“ und die 2 in „Names.2“ ist die DatensatzID des Ausschnitt-Datensatzes.

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

So verwenden Sie eine Abfrage, um mehrere Ausschnitt-Datensätze über den Ausgangsdatensatz zu bearbeiten:

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

Sie können auch den Befehl `-edit` verwenden und 0 als ID des Ausschnitt-Datensatzes angeben, um dem Ausschnitt einen neuen Bezugsdatensatz für einen bestehenden Ausgangsdatensatz hinzuzufügen.

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

Löschen von Ausschnittdatensätzen

Um Ausschnittdatensätze zu löschen, verwenden Sie den Parameter `-delete.related` mit dem Befehl `-edit` anstelle des Befehls `-delete`.

Beispiele

Der folgende URL löscht den Datensatz „1001“ aus der Tabelle „employees“:

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

Aber der folgende URL löscht einen Ausschnitt Datensatz mit einer DatensatzID „3“ aus der Bezugstabelle „Dependents“ mit der AusgangsdatsatzID „1001“.

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

Siehe „-delete.related (Löschen von Ausschnitt Datensätzen, Abfrage-Parameter)“ auf Seite 53.

Abfragen von Ausschnittfeldern

In einer Lösung mit vielen Datensätzen kann das Abfragen und Sortieren von Datensätzen einige Zeit in Anspruch nehmen. Um die anzuzeigende Anzahl von Datensätzen und Zeilen in einer Bezugsmenge einzuschränken, verwenden Sie die Parameter `-relatedsets.filter` und `-relatedsets.max` bei Suchabfragen. Weitere Informationen hierzu finden Sie unter „Abfrage-Parameter `-relatedsets.filter` (Ausschnitt Datensätze filtern)“ auf Seite 59 und „Abfrage-Parameter `-relatedsets.max` (Ausschnitt Datensätze beschränken)“ auf Seite 60.

Erläuterung der Syntax für das Angeben eines Variablenfelds

Syntax für das Angeben eines Variablenfelds:

```
tabellen-name::feld-name(wiederholung-nummer).global
```

Dabei gilt: `global` gibt ein Feld an, das global gespeichert wird. Informationen über `tabellen-name` und `feld-name(wiederholung-nummer)` finden Sie unter „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 46. Informationen über Variablenfelder finden Sie in der [FileMaker Pro Advanced Hilfe](#).

Sie müssen die `.global`-Syntax verwenden, um ein Variablenfeld in einem Abfrage-String zu identifizieren. Die Web Publishing Engine setzt die Parameterwerte für Variablenfelder, bevor sie den Abfrage-Befehl ausführt oder andere Parameterwerte im Abfrage-String setzt. Bei direkten XML-Abfragen werden die globalen Werte direkt nach Ausführung der Abfrage ungültig.

Wenn Sie ein Variablenfeld in einem Abfrage-String nicht mit der `.global`-Syntax identifizieren, berechnet die Web Publishing Engine das Variablenfeld zusammen mit dem restlichen Abfrage-String, ohne zuvor den Wert für das Variablenfeld zu setzen.

Beispiel:

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

Abfrage-Befehlsreferenz

Dieser Abschnitt enthält Informationen über die Abfrage-Befehle, die für XML-Abfragen zur Verfügung stehen.

Abfrage-Befehl `-dbnames` (Datenbanknamen)

Ruft die Namen aller Datenbanken ab, die FileMaker Server bereitstellt und die für Custom Web Publishing mit XML aktiviert sind.

Erforderliche Abfrage-Parameter: (keine)

Beispiel:

So rufen Sie die Datenbanknamen ab:

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

Abfrage-Befehl `-delete` (Datensatz löschen)

Löscht den vom `-recid`-Parameter angegebenen Datensatz.

Erforderliche Abfrage-Parameter: `-db`, `-lay`, `-recid`

Optionaler Abfrage-Parameter: `-script`

Beispiel:

So löschen Sie einen Datensatz:

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

Abfrage-Befehl `-dup` (Datensatz duplizieren)

Dupliziert den vom `-recid`-Parameter angegebenen Datensatz.

Erforderliche Abfrage-Parameter: `-db`, `-lay`, `-recid`

Optionaler Abfrage-Parameter: `-script`

Beispiel:

So duplizieren Sie den angegebenen Datensatz:

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

Abfrage-Befehl `-edit` (Datensatz bearbeiten)

Aktualisiert den vom Parameter `-recid` angegebenen Datensatz und füllt die Felder mit dem Inhalt etwaiger Feldname-Wert-Paare. Der `-recid`-Parameter gibt an, welcher Datensatz bearbeitet werden sollte.

Erforderliche Abfrage-Parameter: `-db`, `-lay`, `-recid`, ein oder mehrere Feldname(n)

Optionaler Abfrage-Parameter: `-modid`, `-script`, Feldname

Hinweis Der Befehl `-edit` kann verwendet werden, um Datensätze in einem Ausschnitt zu bearbeiten. Siehe „Bearbeiten von Datensätzen in einem Ausschnitt“ auf Seite 47.

Beispiel:

So bearbeiten Sie einen Datensatz:

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

Abfrage-Befehle -find, -findall oder -findany (Datensätze suchen)

Sendet eine Suchabfrage mit definierten Kriterien.

Erforderliche Abfrage-Parameter: -db, -lay

Optionale Abfrage-Parameter: -recid, -lop, -op, -max, -skip, -sortorder, -sortfield, -script, -script.prefind, -script.presort, **Feldname**

Beispiele

So suchen Sie einen Datensatz nach Feldname:

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

So suchen Sie einen Datensatz nach DatensatzID:

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

Um alle Datensätze in der Datenbank zu finden, verwenden Sie -findall:

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

Um einen zufällig ermittelten Datensatz zu finden, verwenden Sie -findany:

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

Hinweise

- Die mehrfache Angabe eines Feldnamens in einer einzelnen Abfrage wird nicht unterstützt. FileMaker Server geht alle Werte durch, verwendet aber nur den zuletzt geparsten Wert.
- Wenn Sie den Befehl -findall verwenden, vermeiden Sie Speicherüberlaufprobleme, indem Sie mit dem Abfrage-Parameter -max eine Standard-Maximalzahl von Datensätzen pro Seite angeben.

Abfrage-Befehl -findquery (Compound-Suche)

Sendet eine Suchabfrage mit mehreren Suchdatensätzen und Datensatzausschluss-Abfragen.

Erforderliche Abfrage-Parameter: -db, -lay, -query

Optionale Abfrage-Parameter: -max, -skip, -sortorder, -sortfield, -script, -script.prefind, -script.presort

Beispiel:

Sucht Datensätze mit Hunden oder Katzen, die nicht „Fluffy“ heißen.

```
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
```

Verwenden des Befehls `-findquery` für Compound-Suchen

Eine Anweisung `-findquery` besteht aus vier Teilen in der folgenden Reihenfolge:

- Parameter `-query`
- Die Abfrage-Abfragedeklarationen, bestehend aus den Abfrage-Bezeichnerdeklarationen und Abfragevorgängen
- Das Suchfeld und Wertedefinitionen für jeden Abfrage-Bezeichner
 - Definieren Sie Abfrage-Bezeichner. Ein Abfrage-Bezeichner ist der Buchstabe „q“ gefolgt von einer Zahl. Beispiel: `-q1`
 - Definieren Sie Abfrage-Bezeichner mit dem Parameter. Beispiel:
`-q1.value=fieldvalue`
 - Definieren Sie Abfrage-Bezeichner-Operatoren durch Einschließen als Teil des Ausdrucks `fieldvalue`. So verwenden Sie z. B. einen Stern als Operator „beginnt mit“:
`-q1.value=fieldvalue*`
- Der Befehl `-findquery` am Ende der gesamten Anweisung

Informationen zur Verwendung des Parameters `-query` finden Sie unter „Abfrage-Parameter `-query` (Compound-Suchabfrage)“ auf Seite 57.

Abfrage-Befehl `-layoutnames` (Layoutnamen)

Ruft die Namen aller verfügbaren Layouts für eine angegebene Datenbank ab, die FileMaker Server bereitstellt und die für Custom Web Publishing mit XML aktiviert sind.

Erforderliche Abfrage-Parameter: `-db`

Beispiel:

So rufen Sie die Namen verfügbarer Layouts ab:

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

Abfrage-Befehl `-new` (Neuer Datensatz)

Erzeugt einen neuen Datensatz und fügt in diesen den Inhalt jeglicher Feldnamen/Wert-Paare ein.

Erforderliche Abfrage-Parameter: `-db, -lay`

Optionaler Abfrage-Parameter: ein oder mehrere Feldname(n), `-script`

Hinweis Informationen über das Einfügen neuer Daten für einen Ausschnitt finden Sie unter „Hinzufügen von Datensätzen in einem Ausschnitt“ auf Seite 46.

Beispiel:

So fügen Sie einen neuen Datensatz hinzu:

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

Abfrage-Befehl `-scriptnames` (Scriptnamen)

Ruft die Namen aller verfügbaren Scripts für eine angegebene Datenbank ab, die FileMaker Server bereitstellt und die für Custom Web Publishing mit XML aktiviert sind.

Erforderliche Abfrage-Parameter: `-db`

Beispiel:

So rufen Sie die Namen aller Scripts ab:

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

Abfrage-Befehl `-view` (Layoutinformationen anzeigen)

Wenn die `FMPXMLLAYOUT`-Grammatik angegeben ist: Ruft Layoutinformationen aus einer Datenbank ab und zeigt sie in der `FMPXMLLAYOUT`-Grammatik an. Wenn eine Datengrammatik (`fmresultset` oder `FMPXMLRESULT`) angegeben ist: ruft den Metadaten-Abschnitt des XML-Dokuments und eine leere Datensatzmenge ab.

Erforderliche Abfrage-Parameter: `-db`, `-lay`

Optionaler Abfrage-Parameter: `-script`

Beispiele

So rufen Sie Layoutinformationen ab:

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

So rufen Sie Metadaten-Informationen ab:

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

Abfrage-Parameterreferenz

Dieser Abschnitt enthält Informationen über die Abfrage-Parameter, die für XML-Abfragen zur Verfügung stehen.

Abfrage-Parameter `-db` (Datenbankname)

Gibt die Datenbank an, auf die der Abfrage-Befehl angewendet wird.

Wert: Name der Datenbank ohne etwaige Dateinamenerweiterung.

Hinweis Geben Sie mit dem Namen der Datenbank für den Parameter `-db` keine Dateinamenerweiterung an. Der tatsächliche Datenbank-Dateiname kann optional eine Erweiterung aufweisen, aber Erweiterungen sind als Wert für den Parameter `-db` nicht erlaubt.

Erforderlich für: Alle Abfrage-Befehle mit Ausnahme von `-dbnames`

Beispiel:

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

-delete.related (Löschen von Ausschnittdatensätzen, Abfrage-Parameter)

Löscht einen Datensatz aus einem Ausschnittfeld.

Optional bei: Abfrage-Befehl `-edit`

Erfordert: Ein Bezugstabellenname und eine DatensatzID

Beispiel:

Im Folgenden wird ein Ausschnittdatensatz mit einer DatensatzID „20“ aus der Bezugstabelle „jobtable“ mit der AusgangsdatsatzID „7“ gelöscht.

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

Abfrage-Parameter -field (Containerfeldname)

Gibt den Namen eines Containerfelds zurück.

Erforderlich für: Abfragen von Daten in einem Containerfeld

Siehe „Erläuterung der URL-Syntax für XML-Daten und Containerobjekte“ auf Seite 27.

Abfrage-Parameter „fieldname“ (Nicht-Containerfeldname)

Feldnamen dienen als Kontrollkriterien für den Abfrage-Befehl `-find` oder zum Ändern des Inhalts eines Datensatzes. Wenn ein Abfrage-Befehl oder -Parameter einen Wert für ein Nicht-Containerfeld verlangt, verwenden Sie den Feldnamen ohne den Bindestrich (-) als den Namensteil des Name-Wert-Paars.

Name: Name des Felds in der FileMaker-Datenbank. Wenn das Feld sich nicht in der zugrunde liegenden Tabelle des Layouts befindet, das im Abfrage-String angegeben ist, muss der Feldname vollständig qualifiziert sein.

Es empfiehlt sich nicht, Punkte oder Klammern in Feldnamen zu verwenden. In bestimmten Fällen funktionieren Feldnamen mit Punkten, aber Feldnamen mit den folgenden Ausnahmen können nie verwendet werden:

- Dem Punkt darf keine Zahl folgen. Beispielsweise ist `meinfeld.9` ein ungültiger Dateiname.
- Dem Punkt darf nicht die Textzeichenfolge `op` (die beiden Buchstaben „o“ und „p“) folgen. Beispielsweise ist `meinfeld.op` ein ungültiger Feldname.
- Dem Punkt darf nicht die Textzeichenfolge `global` (das Wort „global“) folgen. Beispielsweise ist `meinfeld.global` ein ungültiger Feldname.

Auf Feldnamen, die eine dieser Ausnahmen enthalten, kann nicht über XML mithilfe einer HTTP-Abfrage zugegriffen werden. Die Verwendung von Punkten in Feldnamen sollte für DatensatzIDs reserviert sein. Siehe dazu „Erläuterung der Syntax für einen voll qualifizierten Feldnamen“ auf Seite 46.

Wert: Geben Sie für die Abfrage-Befehle `-new` und `-edit` den Wert an, den Sie im aktuellen Datensatz in dem Feld speichern wollen. Geben Sie für den Abfrage-Befehl `-find` den Wert an, den Sie im Feld suchen wollen. Wenn Sie den Wert für ein Datums-, Zeit- oder Zeitstempelfeld angeben, müssen Sie den Wert gemäß „fm“-Format für den Feldtyp angeben. Die „fm“-Formate sind `MM/DD/YYYY` für Datum, `hh:mm:ss` für Zeit und `MM/DD/YYYY hh:mm:ss` für Zeitstempel.

Erforderlich für: Abfrage-Befehl `-edit`

Optional bei: Abfrage-Befehlen `-new` und `-find`

Beispiel:

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

Hinweis Die mehrfache Angabe eines Feldnamens in einer einzelnen Abfrage wird nicht unterstützt. FileMaker Server geht alle Werte durch, verwendet aber nur den zuletzt geparsten Wert.

Abfrage-Parameter „fieldname.op“ (Vergleichsoperator)

Gibt den Vergleichsoperator an, der auf den Feldnamen, der dem Operator vorangeht, angewendet werden soll. Vergleichsoperatoren werden mit dem Abfrage-Befehl `-find` verwendet.

Wert: Der zu verwendende Operator. Gültige Operatoren sind:

Schlüsselwort	Äquivalenter Operator in FileMaker Pro Advanced
eq	= Wort
cn	*Wort*
bw	Wort*
ew	*Wort
gt	> Wort
gte	>= Wort
lt	< Wort
lte	<= Wort
neq	ausschließen, Wort

Optional bei: Abfrage-Befehl `-find`

Erfordert: Feldname und Wert

Syntax für das Angeben eines Vergleichsoperators:

```
tabelle-name::feld-name=value&tabelle-name::feld-name.op=op-symbol
```

Dabei gilt:

- `tabelle-name` ist die Tabelle, die das Feld enthält, und ist nur erforderlich, wenn das Feld nicht in der Quelltable des im Abfrage-String angegebenen Layouts ist.
- `op-symbol` ist eines der Schlüsselwörter in der vorherigen Tabelle, wie z. B. `cn`.

Beispiel:

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

Hinweis Das Schlüsselwort `bw` funktioniert nicht mit Datums-, Zeit- oder Zeitstempelzeichenfolgen oder dem Suchoperator für das aktuelle Datum (`///`).

Sie können jeden FileMaker Pro Advanced-Suchoperator verwenden, indem Sie ihn als Teil der Suchkriterien anstelle der Angabe des Operatorschlüsselworts `fieldname.op` angeben. Um z. B. einen Wertebereich mit dem Suchoperator Bereich (. . .) zu suchen, geben Sie kein Operatorschlüsselwort an. Verwenden Sie stattdessen die Zeichen „. . .“ zwischen den Bereichswerten in den Suchkriterien.

Beispiel:

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

Informationen über Operatoren für die Suche nach Text finden Sie in der [FileMaker Pro Advanced Hilfe](#).

Abfrage-Parameter `-lay` (Layout)

Gibt das Datenbanklayout an, das Sie verwenden wollen.

Wert: Name des Layouts

Erforderlich für: Alle Abfrage-Befehle mit Ausnahme von `-dbnames`, `-layoutnames` und `-scriptnames`.

Beispiel:

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

Abfrage-Parameter `-lay.response` (Layout wechseln für Antwort)

Gibt an, dass FileMaker Server beim Verarbeiten einer Abfrage das Layout verwenden soll, das der Parameter `-lay` angibt, und bei der Verarbeitung der XML-Antwort in das Layout wechseln soll, das der Parameter `-lay.response` angibt.

Wenn Sie den Parameter `-lay.response` nicht angeben, verwendet FileMaker Server das vom Parameter `-lay` angegebene Layout für die Verarbeitung der Abfrage und der Antwort.

Sie können den Parameter `-lay.response` für XML-Abfragen verwenden.

Wert: Name des Layouts

Optional bei: Alle Abfrage-Befehle mit Ausnahme von `-dbnames`, `-layoutnames` und `-scriptnames`.

Beispiel:

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

Abfrage-Parameter `-lop` (Logischer Operator)

Gibt an, wie die Suchkriterien im Abfrage-Befehl `-find` als „and“- oder „or“-Suche kombiniert werden.

Wert: `and` oder `or`

Wenn der Abfrage-Parameter `-lop` nicht angegeben wird, verwendet der Abfrage-Befehl `-find` den Wert „and“.

Optional bei: Abfrage-Befehl `-find`

Hinweis Nicht unterstützt vom Abfrage-Befehl `-findquery`.

Beispiel:

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

Abfrage-Parameter `-max` (Maximum Datensätze)

Gibt die maximale Anzahl an Datensätzen an, die zurückgegeben werden soll.

Wert: Eine Zahl oder `-` für die Rückgabe aller Datensätze – der Wert `all`. Wenn `-max` nicht angegeben wird, werden alle Datensätze zurückgegeben.

Optional bei: Abfrage-Befehle `-find`, `-findall` und `-findquery`

Hinweis Der Abfrage-Parameter `-max` wirkt sich nicht auf die für Ausschnittdatensätze zurückgegebenen Werte aus. Um die Anzahl der für Ausschnittdatensätze zurückgegebenen Zeilen einzuschränken, siehe „Abfrage-Parameter `-relatedsets.max` (Ausschnittdatensätze beschränken)“ auf Seite 60.

Beispiel:

```
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
```

Abfrage-Parameter `-modid` (ÄnderungsID)

Die ÄnderungsID ist ein inkrementeller Zähler, der die aktuelle Version eines Datensatzes angibt. Indem Sie mit dem Abfrage-Befehl `-edit` eine ÄnderungsID angeben, können Sie sicherstellen, dass Sie die aktuelle Version eines Datensatzes bearbeiten. Wenn der angegebene Wert der ÄnderungsID nicht dem aktuellen Wert der ÄnderungsID in der Datenbank entspricht, wird der Abfrage-Befehl `-edit` nicht zugelassen und ein Fehlercode wird zurückgegeben.

Wert: Eine ÄnderungsID, die die aktuelle Version eines Datensatzes in einer FileMaker-Datenbank eindeutig identifiziert.

Optional bei: Abfrage-Befehl `-edit`

Erfordert: `-recid`-Parameter

Beispiel:

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

Abfrage-Parameter `-query` (Compound-Suchabfrage)

Gibt die Abfrage-Namen und Suchkriterien für eine Compound-Suchabfrage an. Siehe „Abfrage-Befehl `-findquery` (Compound-Suche)“ auf Seite 50.

Wert: Ein Abfrage-Ausdruck.

Erforderlich für: Abfrage-Befehl `-findquery`

Die Syntax für eine Compound-Suchabfrage ist:

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

Dabei gilt:

`<request-declarations>` sind zwei oder mehr Abfragedeklarationen.

- Jede Abfragedeklaration setzt sich aus einem oder mehreren Abfrage-Bezeichnern zusammen, die durch Kommata getrennt und in Klammern eingeschlossen sind. Ein Abfrage-Bezeichner ist der Buchstabe „q“ gefolgt von einer Zahl. Beispiel: `q1`
- Eingeschlossen in Klammern fungieren mehrere Abfragen als logische UND-Suchen, die die Ergebnismenge einengen. Zum Beispiel gibt `(q1, q2)` Datensätze zurück, die `q1` und `q2` entsprechen.

Hinweis Es empfiehlt sich nicht, die gleichen Felder für mehrere `q`-Variablen in den gleichen „and“-Suchkriterien zu verwenden.

- Wie bei FileMaker Pro Advanced kann jede Abfrage eine Such- oder Ausschlussabfrage sein. Eine Suchabfrage fügt die entsprechenden Datensätze der Ergebnismenge hinzu. Eine Ausschlussabfrage entfernt die entsprechenden Datensätze aus der Ergebnismenge. Standard ist eine Suchabfrage. Setzen Sie für eine Ausschlussabfrage ein Ausrufezeichen (!) vor die öffnende Klammer.

Beispiel:

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

`q1` ist eine Suchabfrage, `q2` ist eine Ausschlussabfrage, da ihr ein Ausrufezeichen vorangestellt ist.

- Abfragen werden durch Semikolon getrennt. Mehrere Suchabfragen fungieren als logische ODER-Suchen, die die Ergebnismenge erweitern. Zum Beispiel gibt `(q1);(q2)` Datensätze zurück, die `q1` oder `q2` entsprechen. Ausschlussabfragen fungieren nicht als logische ODER-Suchen, da Ausschlussabfragen Datensätze aus der Ergebnismenge entfernen.
- Abfragen werden in der angegebenen Reihenfolge ausgeführt. Die Ergebnismenge umfasst die Ergebnisse aus der gesamten zusammengesetzten Suchabfrage.

`<request-definitions>` ist eine Abfragedefinition für jede Abfragedeklaration. Jede Abfragedefinition besteht aus Suchfeld und Wertdefinition. Ein Minuszeichen (-) beginnt die Abfragedefinition.

Syntax:

```
-<query-id>=<Feldname>&-<query-id>.value=<Wert>
```

Beispiele

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

Sucht Datensätze mit grauen Katzen, die nicht „Fluffy“ heißen.

```
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
```

Abfrage-Parameter `-recid` (DatensatzID)

Gibt den Datensatz an, der verarbeitet werden soll. Wird hauptsächlich mit den Abfrage-Befehlen `-edit` und `-delete` verwendet. Vom Befehl `-view` verwendet, um Bezugswertlistendaten in der Grammatik FMPXMLLAYOUT abzurufen.

Wert: Eine DatensatzID, die einen Datensatz in einer FileMaker-Datenbank eindeutig angibt.

Erforderlich für: Abfrage-Befehle `-edit`, `-delete` und `-dup`

Optional bei: Befehlen `-find query` und `-view`

Beispiele

```
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
```

Abfrage-Parameter `-relatedsets.filter` (Ausschnittdatensätze filtern)

Gibt an, ob die Ausschnittdatensätze im Ergebnis für diese Abfrage eingeschränkt werden sollen.

Wert: `layout` oder `none`.

- Wenn `-relatedsets.filter` auf `layout` eingestellt ist, wird die Einstellung **Anfangszeile** im FileMaker Pro Advanced-Dialogfeld „Ausschnitteinstellung“ beachtet.
 - Wenn die Einstellung **Vertikaler Rollbalken** im Dialogfeld „Ausschnitteinstellungen“ aktiviert ist, verwenden Sie die Option `-relatedsets.max`, um die maximale Anzahl an zurückzugebenden Datensätzen festzulegen. Siehe „Abfrage-Parameter `-relatedsets.max` (Ausschnittdatensätze beschränken)“ weiter unten.
 - Wenn die Einstellung **Vertikaler Rollbalken** deaktiviert oder die Option `-relatedsets.max` nicht verwendet wird, bestimmt die Einstellung **Anzahl der Zeilen** im Dialogfeld „Ausschnitteinstellungen“ die Anzahl der zurückzugebenden Ausschnittdatensätze.
- Der Standardwert ist `none`, wenn dieser Parameter nicht angegeben ist. Wenn `-relatedsets.filter` auf `none` eingestellt ist, gibt die Web Publishing Engine alle Datensätze im Ausschnitt zurück. Die Werte für **Anfangszeile** und **Anzahl der Zeilen**, die im Dialogfeld „Ausschnitteinstellungen“ angegeben wurden, werden ignoriert.

Hinweise

- Der Parameter `-relatedsets.filter` hat keine Auswirkung darauf, wie Ausschnittdatensätze in XML-Abfragen sortiert werden. Die in FileMaker Pro Advanced angegebene Sortierung wird beachtet, unabhängig davon, ob der Parameter `-relatedsets.filter` auf `layout` oder `none` gesetzt ist.
- Die Einstellung **Ausschnittdatensätze filtern** im Dialogfeld „Ausschnitteinstellungen“ wird für XML-Abfragen nicht unterstützt. Etwaige Formeln, die für die Einstellung **Ausschnittdatensätze filtern** angegeben sind, werden ignoriert.

Optional bei: `-find`, `-edit`, `-new`, `-dup` und `-findquery`.

Beispiele

```
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
```

Abfrage-Parameter `-relatedsets.max` (Ausschnittdatensätze beschränken)

Gibt die maximale Anzahl an Ausschnittdatensätzen an, die in den Ergebnissen für diese Abfrage zurückgegeben werden.

Wert: eine Ganzzahl oder `all`.

- Der Parameter `-relatedsets.max` wird nur beachtet, wenn die Einstellung **Vertikaler Rollbalken** im FileMaker Pro Advanced-Dialogfeld „Ausschnitteinstellung“ und der Parameter `-relatedsets.filter` auf `layout` gesetzt sind.
 - Wenn der Parameter `-relatedsets.max` eine Ganzzahl angibt, gibt die Web Publishing Engine die Anzahl an Ausschnittdatensätzen beginnend mit der Anfangszeile zurück.
 - Wenn der Parameter `-relatedsets.max` `all` angibt, gibt die Web Publishing Engine alle Ausschnittdatensätze zurück.

Hinweis Informationen über das Filtern von Ausschnittdatensätzen finden Sie weiter oben unter „Abfrage-Parameter `-relatedsets.filter` (Ausschnittdatensätze filtern)“.

Optional bei: `-find`, `-edit`, `-new`, `-dup` und `-findquery`.

Beispiele

```
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
```

Abfrage-Parameter `-script` (Script)

Gibt das FileMaker-Script an, das nach Ausführung des Abfrage-Befehls und der Sortierung ablaufen soll. Siehe „Erläuterung der Verarbeitung einer XML-Abfrage“ auf Seite 42.

Wert: Scriptname

Optional bei: Alle Abfrage-Befehle mit Ausnahme von `-dbnames`, `-layoutnames` und `-scriptnames`.

Beispiel:

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

Abfrage-Parameter `-script.param` (Parameter an Script übergeben)

Übergibt einen Parameter an das durch `-script` angegebene FileMaker-Script.

Wert: Ein einzelner Textparameter.

- Um mehrere Parameter zu übergeben, können Sie einen String erstellen, der die Parameter begrenzt, und Ihr Script die einzelnen Parameter auslesen lassen. Übergeben Sie z. B. `param1|param2|param3` als Liste mit dem Zeichen `|`, das so kodiert ist:
`param1%7Cparam2%7Cparam3`
- Um den Textparameter als Wert zu behandeln, der kein Text ist, kann Ihr Script den Textwert konvertieren. Um z. B. den Textwert in eine Zahl zu konvertieren, könnte Ihr Script Folgendes beinhalten: `LiesAlsZahl (Hole (ScriptParam))`
- Wenn Ihre Abfrage `-script.param` ohne `-script` enthält, wird `-script.param` ignoriert.
- Wenn Ihre Abfrage mehr als einen `-script.param` enthält, verwendet die Web Publishing Engine den letzten geparsten Wert.

Optional bei: `-script`

Beispiel:

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

Abfrage-Parameter `-script.prefind` (Script vor Suchen)

Gibt das FileMaker-Script an, das vor dem Suchen und Sortieren von Datensätzen (falls angegeben) während der Verarbeitung des Abfrage-Befehls `-find` ausgeführt wird.

Wert: Scriptname

Optional bei: Alle Abfrage-Befehle mit Ausnahme von `-dbnames`, `-layoutnames` und `-scriptnames`.

Beispiel:

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

Abfrage-Parameter `-script.prefind.param` (Parameter an Script vor Suchen übergeben)

Übergibt einen Parameter an das durch `-script.prefind` angegebene FileMaker-Script.

Wert: Ein einzelner Textparameter.

- Um mehrere Parameter zu übergeben, können Sie einen String erstellen, der die Parameter begrenzt, und Ihr Script die einzelnen Parameter auslesen lassen. Übergeben Sie z. B. `param1|param2|param3` als Liste mit dem Zeichen `|`, das so kodiert ist:
`param1%7Cparam2%7Cparam3`
- Um den Textparameter als Wert zu behandeln, der kein Text ist, kann Ihr Script den Textwert konvertieren. Um z. B. den Textwert in eine Zahl zu konvertieren, könnte Ihr Script Folgendes beinhalten: `LiesAlsZahl (Hole (ScriptParam))`
- Wenn Ihre Abfrage `-script.prefind.param` ohne `-script.prefind` enthält, wird `-script.prefind.param` ignoriert.
- Wenn Ihre Abfrage mehr als einen `-script.prefind.param` enthält, verwendet die Web Publishing Engine den letzten geparsten Wert.

Optional bei: `-script.prefind`

Beispiel:

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

Abfrage-Parameter `-script.presort` (Script vor Sortieren)

Gibt das FileMaker-Script an, das nach dem Suchen von Datensätzen (falls angegeben) und vor dem Sortieren der Datensätze während der Verarbeitung des Abfrage-Befehls `-find` ausgeführt wird.

Optional bei: Alle Abfrage-Befehle mit Ausnahme von `-dbnames`, `-layoutnames` und `-scriptnames`.

Beispiel:

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

Abfrage-Parameter `-script.presort.param` (Parameter an Script vor Sortieren übergeben)

Übergibt einen Parameter an das durch `-script.presort` angegebene FileMaker-Script.

Wert: Ein einzelner Textparameter.

- Um mehrere Parameter zu übergeben, können Sie einen String erstellen, der die Parameter begrenzt, und Ihr Script die einzelnen Parameter auslesen lassen. Übergeben Sie z. B. `param1|param2|param3` als Liste mit dem Zeichen `|`, das so kodiert ist:
`param1%7Cparam2%7Cparam3`

- Um den Textparameter als Wert zu behandeln, der kein Text ist, kann Ihr Script den Textwert konvertieren. Um z. B. den Textwert in eine Zahl zu konvertieren, könnte Ihr Script Folgendes beinhalten: `LiesAlsZahl (Hole(ScriptParam))`
- Wenn Ihre Abfrage `-script.presort.param` ohne `-script.presort` enthält, wird `-script.presort.param` ignoriert.
- Wenn Ihre Abfrage mehr als einen `-script.presort.param` enthält, verwendet die Web Publishing Engine den letzten gepassten Wert.

Optional bei: `-script.presort`

Beispiel:

```
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
```

Abfrage-Parameter `-skip` (Datensätze überspringen)

Gibt an, wie viele Datensätze in der Ergebnismenge übersprungen werden sollen.

Wert: Eine Zahl. Wenn der Wert größer als die Anzahl der Datensätze in der Ergebnismenge ist, wird kein Datensatz angezeigt. Der Standardwert beträgt 0.

Optional bei: Abfrage-Befehl `-find`

Beispiel:

Die ersten 10 Datensätze in der Ergebnismenge werden übersprungen, die Datensätze 11 bis 15 werden zurückgegeben:

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

Abfrage-Parameter `-sortfield` (Sortierfeld)

Gibt das Feld an, das für das Sortieren verwendet wird.

Wert: Feldname

Optional bei: Abfrage-Befehlen `-find-` oder `-findall`

Der Abfrage-Parameter `-sortfield` kann mehrfach verwendet werden, um mehrere Feldsortierungen vorzunehmen. Syntax für die Rangfolge der Sortierfelder:

```
-sortfield.rang-nummer=voll-qualifizierter-feldname
```

Dabei gilt: `rang-nummer` im Abfrage-Parameter `-sortfield.rang-nummer` ist eine Zahl, die die Rangfolge für mehrere Sortierfelder angibt. Der Wert für `precedence-number`:

- muss bei 1 beginnen.
- muss sequenziell heraufgesetzt werden.
- darf nicht größer als 9 sein.

Beispiel:

Zuerst wird das Feld „dept“ sortiert und dann das Feld „rating“. Beide Felder werden aufsteigend sortiert, da der Abfrageparameter `-sortorder` nicht angegeben wurde.

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

Abfrage-Parameter `-sortorder` (Sortierfolge)

Gibt die Richtung einer Sortierung an.

Wert: Die Sortierfolge. Gültige Sortierfolgen (wobei `<wertelisten-name>` der Name einer Werteliste wie `Spezial` ist):

Schlüsselwort	Äquivalenter Operator in FileMaker Pro Advanced
<code>ascend</code>	Sortieren von a bis z, -10 bis -10
<code>descend</code>	Sortieren von z bis a, 10 bis -10
<code><wertelisten-name></code>	Sortieren nach der angegebenen Werteliste für das Feld im Layout

Optional bei: Abfrage-Befehlen `-find-` oder `-findall`

Erfordert: `-sortfield` (Abfrage-Parameter)

Der Abfrage-Parameter `-sortorder` kann mit dem Abfrage-Parameter `-sortfield` verwendet werden, um die Sortierfolge mehrerer Sortierfelder anzugeben. Syntax für die Sortierfolge eines Sortierfelds:

```
-sortorder.rang-nummer=sortier-methode
```

Dabei gilt:

- `rang-nummer` im Abfrage-Parameter `-sortorder.rang-nummer` ist eine Zahl von 1 bis 9, die den Abfrage-Parameter `-sortfield` angibt, auf den sich der Abfrage-Parameter `-sortorder` bezieht.
- `sortier-methode` ist eines der Schlüsselwörter in obiger Tabelle für die Festlegung der Sortierfolge, z. B. „`ascend`“.

Beispiel:

Die Sortierfolge für das Sortierfeld mit der höchsten Rangnummer (`dept`) ist „`ascend`“ und für das Sortierfeld mit der zweithöchsten Rangnummer (`rating`) „`descend`“. Die `precedence-number 2` in `-sortorder.2` gibt an, dass der Abfrage-Parameter `-sortorder.2=descend` sich auf den Abfrage-Parameter `-sortfield.2=rating` bezieht.

```
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
```

Hinweis Ein Sortierfeld, für das der Abfrage-Parameter `-sortorder` nicht angegeben ist, verwendet standardmäßig aufsteigende Sortierung.

Kapitel 6

Erläuterung von Custom Web Publishing mit PHP

Custom Web Publishing mit PHP ermöglicht die Verwendung der Scriptsprache PHP, um Daten aus FileMaker-Datenbanken in Ihre angepassten Webseiten-Layouts zu integrieren. Custom Web Publishing mit PHP stellt FileMaker API für PHP bereit, bei dem es sich um eine PHP-Klasse handelt, die von FileMaker erzeugt wird und auf Datenbanken zugreift, die von FileMaker Server bereitgestellt werden. Diese PHP-Klasse stellt eine Verbindung mit der FileMaker Server Web Publishing Engine her und macht Daten für die PHP Engine Ihres Web-Servers verfügbar.

Wichtige Funktionen in Custom Web Publishing mit PHP

- Erstellen Sie Webanwendungen, die die Open-Source-Scriptsprache PHP verwenden. Verwenden Sie die von FileMaker Server unterstützte Version PHP oder Ihre eigene Version von PHP. (Wenn Sie Ihre eigene Version von PHP verwenden, siehe „Manuelle Installation von FileMaker API für PHP“ auf Seite 67.)
- Stellen Sie Datenbanken mit FileMaker Server bereit. FileMaker Pro Advanced ist für Custom Web Publishing nicht erforderlich, da FileMaker Server die Datenbanken bereitstellt.
- Schreiben Sie PHP-Code, mit dem Datensätze in einer bereitgestellten FileMaker-Datenbank erstellt, gelöscht, bearbeitet und dupliziert werden können. Mit Ihrem Code können Feld- und Datensatzüberprüfungen durchgeführt werden, bevor Änderungen in der bereitgestellten Datenbank bestätigt werden.
- Schreiben Sie PHP-Code, der auf Layouts, Ausschnitte, Wertelisten und Bezugfelder zugreift. Wie in FileMaker Pro Advanced basiert der Zugriff auf Daten, Layouts und Felder auf den Benutzerkontoeinstellungen, die in den Zugriffsrechten der Datenbank definiert sind. Die Web Publishing Engine unterstützt auch mehrere andere Sicherheitsverbesserungen. Siehe „Schützen veröffentlichter Datenbanken“ auf Seite 14.
- Schreiben Sie PHP-Code, der komplexe Scripts mit mehreren Schritten ausführt. FileMaker unterstützt viele 70 Stepschritte in Custom Web Publishing. Siehe „FileMaker-Scripts und Custom Web Publishing“ auf Seite 18.
- Schreiben Sie PHP-Code, der komplexe Suchabfragen ausführt.

Anforderungen für Custom Web Publishing

Dieser Abschnitt beschreibt, was für die Entwicklung einer Custom Web Publishing-Lösung mit PHP erforderlich ist, was Webbenutzer benötigen, um auf eine Custom Web Publishing-Lösung zuzugreifen und welche Auswirkungen die Bereitstellung einer Web Publishing-Lösung auf Ihren Server haben könnte.

Voraussetzungen für das Veröffentlichen einer Datenbank mit Custom Web Publishing

Um Datenbanken mithilfe von Custom Web Publishing mit PHP zu veröffentlichen, benötigen Sie:

- einen FileMaker Server-Einsatz, der sich aus drei Komponenten zusammensetzt.
 - einen Web-Server, entweder Microsoft IIS (Windows) oder Apache (macOS); das FileMaker Web-Server-Modul ist auf dem Web-Server installiert.
 - die FileMaker Web Publishing Engine.
 - den FileMaker Datenbank-Server.
- eine oder mehrere FileMaker Pro-Advanced Datenbanken, die FileMaker Server bereitstellt.
- IP-Adresse oder Domänenname des Hosts, auf dem der Web-Server läuft.
- einen beliebigen Webbrowser und Zugriff auf den Web-Server, um Ihre Custom Web Publishing-Lösung zu entwickeln und zu testen.
- FileMaker Server installiert die unterstützte Version von PHP oder Sie können Ihre eigene Version verwenden.
 - Angaben zur geforderten Mindestversion von PHP finden Sie unter [FileMaker Server - Technische Daten](#).
 - Weitere Informationen zu PHP finden Sie unter [php.net](#).
 - Die auf dem Web-Server installierte PHP-Version muss cURL-Funktionen (Client URL Library) unterstützen. Weitere Informationen zu cURL finden Sie unter [php.net/curl](#).

Wichtig Wenn Sie FileMaker Server installieren, wird die unterstützte Version von PHP nicht im macOS Server Admin-Tool angezeigt. Dies soll auch nicht angezeigt werden. Wenn Sie das macOS Server Admin-Tool für die Aktivierung von PHP verwenden, deaktivieren Sie die von FileMaker Server unterstützte Version von PHP und aktivieren Sie Ihre eigene PHP-Version.

Weitere Informationen finden Sie im Handbuch [FileMaker Server Installation und Konfiguration](#).

Voraussetzungen für Web-Benutzer zum Zugriff auf eine Custom Web Publishing-Lösung

Für den Zugriff auf eine Custom Web Publishing-Lösung mit PHP benötigen Webbenutzer:

- einen Webbrowser
- Zugang zum Internet oder zu einem Intranet und zum Web-Server
- IP-Adresse oder Domänenname des Hosts, auf dem der Web-Server läuft

Wenn die Datenbank passwortgeschützt ist, müssen Webbenutzer auch einen Benutzernamen und ein Passwort für ein Datenbankkonto eingeben.

Verbindung zum Internet oder zu einem Intranet

Wenn Sie Datenbanken im Internet oder in einem Intranet veröffentlichen, muss auf dem Host-Computer FileMaker Server laufen und die freizugebenden Datenbanken müssen bereitgestellt und verfügbar sein. Zusätzlich:

- Veröffentlichen Sie Ihre Datenbank auf einem Computer, der über eine ständige Verbindung zum Internet bzw. Intranet verfügt. Sie können Datenbanken zwar ohne ständige Verbindung veröffentlichen, aber sie stehen Webbenutzern nur zur Verfügung, wenn Ihr Computer mit dem Internet bzw. Intranet verbunden ist.
- Der Host-Computer für den Web-Server, der Teil des FileMaker Server-Einsatzes ist, muss über eine eigene statische (permanente) IP-Adresse oder einen Domännennamen verfügen. Wenn Sie die Verbindung zum Internet über einen Internet-Anbieter (ISP) herstellen, könnte Ihre IP-Adresse dynamisch zugewiesen werden (d. h., sie ändert sich bei jeder Verbindung). Mit einer dynamischen IP-Adresse ist es für Webbenutzer schwieriger, Ihre Datenbanken zu finden. Wenn Sie sich über Ihre Zugangsart nicht sicher sind, wenden Sie sich an Ihren ISP oder Netzwerkadministrator.

Manuelle Installation von FileMaker API für PHP

Wenn Sie FileMaker Server installieren, wird die von FileMaker unterstützte Version von PHP installiert. Wenn eine andere PHP-Engine installiert und konfiguriert ist und Sie nur FileMaker API für PHP hinzufügen möchten, installieren Sie die Klasse FileMaker API für PHP manuell, um sie für Ihre PHP-Skripts zur Verfügung zu stellen. Verwenden Sie CLI, um die von FileMaker unterstützte Version von PHP zu deaktivieren. Siehe CLI-Hilfe.

Wenn Sie Ihre eigene PHP Engine verwenden, stellen Sie sicher, dass Sie die folgenden Konfigurationsaufgaben für Ihre Version der PHP Engine durchführen:

- Aktivieren Sie das cURL-Modul in php.ini.
- Geben Sie den Speicherort des FileMaker API für PHP in der Variable `include_path` in php.ini an.
- Wenn Sie auf Datenbanken zugreifen, die Datums- und Zeitwerte enthalten, installieren Sie das [Pear-Date-Paket](#).

Hinweis Angaben zur geforderten Mindestversion von PHP finden Sie unter [FileMaker Server - Technische Daten](#). Verwenden Sie für optimale Ergebnisse die entsprechende PHP-Version.

Bereitstellung von FileMaker API für PHP für Ihre PHP-Skripts

Bei der Installation von FileMaker Server wurde das FileMaker API für PHP-Paket als .zip-Datei an folgendem Ort gespeichert:

- Für IIS (Windows):
`[Laufwerk]:\Programme\FileMaker\FileMaker Server\Web Publishing\FM_API_for_PHP_Standalone.zip`
wobei sich [Laufwerk] auf das Laufwerk bezieht, auf dem die Web-Server-Komponente Ihres FileMaker-Server-Einsatzes gespeichert ist.
- Für Apache (macOS):
`/Library/FileMaker Server/Web Publishing/FM_API_for_PHP_Standalone.zip`

Die Datei FM_API_for_PHP_Standalone.zip enthält die Datei „FileMaker.php“ und den Ordner „FileMaker“. Entzippen Sie die Datei und kopieren Sie die Datei „FileMaker.php“ und den Ordner „FileMaker“ in einen der folgenden Orte:

- den Ordner, in dem Ihre PHP-Skripts gespeichert sind.
 - Für IIS (Windows) durch HTTP oder HTTPS:
[Laufwerk]:\Programme\FileMaker\FileMaker Server\HTTPServer\Conf
wobei [Laufwerk] das Laufwerk bezeichnet, auf dem sich die Web Publishing Engine-Komponente Ihres FileMaker Server-Einsatzes befindet.
 - Für Apache (macOS) durch HTTP:
/Library/FileMaker Server/HTTPServer/htdocs
- Für Apache (macOS) durch HTTPS:
/Library/FileMaker Server/HTTPServer/htdocs/httpsRoot

Hinweis Wenn Sie ein eigenes SSL-Zertifikat importieren, verwenden Datenbank-Server-Client-Verbindungen SSL und HTTP-Verbindungen werden über HTTPS geroutet. Verwenden Sie mit einem eigenen SSL-Zertifikat auf Ihrem Server das HTTPS-Verzeichnis für die Bereitstellung von PHP-Site-Dateien.

- eines der include_path-Verzeichnisse in Ihrer PHP-Installation. Der Standardpfad für macOS ist /usr/lib/php.

Die nächsten Schritte

- Verwenden Sie die CLI, um Custom Web Publishing zu aktivieren. Weitere Informationen finden Sie in der [FileMaker Server Hilfe](#).
- Öffnen Sie in FileMaker Pro Advanced jede FileMaker-Datenbank, die Sie veröffentlichen wollen, und stellen Sie sicher, dass in der Datenbank die geeigneten erweiterten Zugriffsrechte für Custom Web Publishing aktiviert sind. Siehe „Aktivieren von Custom Web Publishing in einer Datenbank“ auf Seite 13.
- Wie Sie mithilfe von FileMaker API für PHP auf Daten in FileMaker-Datenbanken zugreifen, erfahren Sie in Kapitel 8 „Verwendung von FileMaker API für PHP“.

Kapitel 7

Übersicht über Custom Web Publishing mit PHP

Das FileMaker API für PHP unterstützt Sie bei der Integration von Daten aus FileMaker Pro Advanced-Datenbanken in PHP-Lösungen. Dieses Kapitel beschreibt, wie PHP mit der FileMaker Server Custom Web Publishing Engine funktioniert. Weitere Informationen über das FileMaker API für PHP finden Sie in Kapitel 8 „Verwendung von FileMaker API für PHP“.

Funktionsweise der Web Publishing Engine mit PHP-Lösungen

FileMaker Server besteht aus drei Komponenten: dem Web-Server, der Web Publishing Engine und dem Datenbank-Server. Weitere Informationen finden Sie im Handbuch [FileMaker Server Installation und Konfiguration](#). Um PHP-Lösungen zu unterstützen, wird eine PHP Engine zusammen mit dem Web-Server auf dem Mastercomputer installiert. FileMaker Server stellt eine PHP-Lösung bereit, wenn Sie die PHP-Dateien auf dem Web-Server des Mastercomputers ablegen.

- Wenn ein Webbenutzer eine PHP-Lösung öffnet, leitet der Web-Server die Anforderung an die PHP-Engine weiter, die den PHP-Code verarbeitet.
- Wenn der PHP-Code Aufrufe für das FileMaker API für PHP enthält, werden diese Aufrufe interpretiert und als Abfragen an die Web Publishing Engine gesendet.
- Die Web Publishing Engine fordert Daten von den Datenbanken an, die der Datenbank-Server bereitstellt.
- Der Datenbank-Server sendet die angeforderten Daten an die Web Publishing Engine.
- Die Web Publishing Engine sendet die Daten als Antwort auf den API-Aufruf an die PHP-Engine auf dem Web-Server.
- Die PHP-Lösung verarbeitet die Daten und zeigt sie dem Webbenutzer an.

Allgemeine Schritte bei Custom Web Publishing mit PHP

1. Verwenden Sie FileMaker Pro Advanced, um Custom Web Publishing für eine Datenbank zu aktivieren. Siehe Kapitel 2 „Vorbereiten von Datenbanken für Custom Web Publishing“.

Hinweis Stellen Sie sicher, dass Sie entsprechende FileMaker-Datenbankberechtigungen verwenden, wenn Sie PHP-Lösungen entwickeln, die an den Endbenutzer gegeben werden. Andernfalls haben Sie in der FileMaker-Datenbank eventuell Zugriff auf Layouts und Funktionen, die dann dem Endbenutzer nicht zur Verfügung stehen und damit zu inkonsistentem Verhalten führen.

2. Verwenden Sie die FileMaker Server CLI, um Custom Web Publishing mit PHP zu aktivieren. Weitere Informationen finden Sie in der [FileMaker Server Hilfe](#).

3. Bestätigen Sie in Admin Console, dass für jede FileMaker-Datenbank, die Sie veröffentlichen, das erweiterte Zugriffsrecht `fmphp` aktiviert ist.
 - Klicken Sie in Admin Console auf die Seite **Datenbanken**.
 - Wählen Sie **Erweiterte Zugriffsrechte einblenden** aus der Menüauswahl neben **Alle Datenbanken**. Der Text **FMPHP** wird für Datenbanken angezeigt, bei denen das erweiterte Zugriffsrecht `fmphp` aktiviert ist.
4. Verwenden Sie PHP-Authoring-Tools, um Ihre PHP-Lösung zu erstellen, und integrieren Sie die FileMaker API-Funktionen in Ihren PHP-Code, um auf Ihre FileMaker-Daten zuzugreifen. Siehe Kapitel 8 „Verwendung von FileMaker API für PHP“.
5. Kopieren oder verschieben Sie Ihre Website-Verzeichnisstruktur und -Dateien in folgenden Ordner auf dem Web-Server des Mastercomputers:
 - Für IIS (Windows) durch HTTP oder HTTPS:
`[Laufwerk]:\Programme\FileMaker\FileMaker Server\HTTPServer\Conf`, wobei `[Laufwerk]` sich auf das Laufwerk auf dem Mastercomputer bezieht, auf dem sich Ihr FileMaker-Server-Einsatz befindet.
 - Für Apache (macOS) durch HTTP:
`/Library/FileMaker Server/HTTPServer/htdocs`
 - Für Apache (macOS) durch HTTPS:
`/Library/FileMaker Server/HTTPServer/htdocs/httpsRoot`

Hinweis Wenn Sie ein eigenes SSL-Zertifikat importieren, verwenden Datenbank-Server-Client-Verbindungen SSL und HTTP-Verbindungen werden über HTTPS geroutet. Verwenden Sie mit einem eigenen SSL-Zertifikat das HTTPS-Verzeichnis für die Bereitstellung von PHP-Site-Dateien.

6. Wenn ein Datenbank-Containerfeld einen Dateiverweis anstelle der eigentlichen Datei speichert, muss das referenzierte Containerobjekt im FileMaker Pro Advanced-Webordner gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird. Kopieren oder verschieben Sie das Objekt in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software.
Siehe „Veröffentlichen des Inhalts von Containerfeldern im Web“ auf Seite 16.
7. Richten Sie unbedingt Sicherheitsmechanismen für Ihre Site oder Ihr Programm ein.
8. Testen Sie Ihre Site mit den Konten und Zugriffsrechten, die für Webbenutzer definiert sind.
9. Geben Sie die Site frei und informieren Sie Benutzer. Der vom Webbenutzer eingegebene URL weist folgendes Format auf:
`http://<server>/<site_path>`
 - `<server>` ist dabei der Rechner, auf dem sich FileMaker Server befindet.
 - `<site_path>` ist der relative Pfad zur Homepage Ihrer Site und ergibt sich durch die unter 5 oben festgelegte Verzeichnisstruktur.

Beispiel:

Wenn Ihr Web-Server 192.168.123.101 ist und sich die Homepage Ihrer Website auf dem Web-Server unter `c:\inetpub\wwwroot\customers\index.php` befindet, lautet der entsprechende URL:

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

Hinweis PHP verwendet die Kodierung Latin-1 (ISO-8859-1). FileMaker Server gibt Unicode- (UTF-8)-Daten zurück. Verwenden Sie die CLI, um die Standardzeichenkodierung anzugeben. Siehe CLI-Hilfe. Für PHP-Sites können Sie entweder UTF-8 oder ISO-8859-1, UTF-8 angeben. Geben Sie die gleiche Einstellung für das `charset`-Attribut im `<HEAD>`-Bereich der PHP-Dateien Ihrer Site ein.

Weitere Informationen zu Einsatz und Verwendung einer PHP-Lösung finden Sie in Kapitel 9 „Einstellen, Testen und Überwachen einer Site“.

Kapitel 8

Verwendung von FileMaker API für PHP

FileMaker API für PHP implementiert eine PHP-Klasse, die Klasse FileMaker, die eine objektorientierte Schnittstelle für FileMaker-Datenbanken bietet. Mithilfe des FileMaker API für PHP kann sowohl auf Daten als auch auf Logik, die in FileMaker Pro Advanced-Datenbanken gespeichert sind, zugegriffen werden. Diese können auch im Internet veröffentlicht oder an andere Anwendungen exportiert werden.

Mit FileMaker API für PHP kann PHP-Code die Funktionen ausführen, die bereits in FileMaker Pro Advanced-Datenbanken verfügbar sind:

- Datensätze erstellen, löschen, bearbeiten und duplizieren
- Suchabfragen ausführen
- Feld- und Datensatzüberprüfungen ausführen
- Layouts verwenden
- FileMaker-Scripts ausführen
- Ausschnitte und Bezugsdatensätze anzeigen
- Wertelisten verwenden

Dieses Kapitel beschreibt die Verwendung der Objekte und Methoden der Klasse FileMaker für die Integration dieser allgemeinen Funktionen in eine PHP-Lösung. FileMaker API für PHP wird in diesem Kapitel nicht erschöpfend behandelt, vielmehr werden die wichtigsten Objekte und Methoden vorgestellt.

Weitere Informationen

Weitere Informationen zu FileMaker API für PHP erhalten Sie in folgenden Ressourcen.

Falls Sie bereits eine PHP-Engine installiert und konfiguriert haben und nur FileMaker API für PHP hinzufügen möchten, finden Sie weitere Informationen unter „Manuelle Installation von FileMaker API für PHP“ auf Seite 67.

Referenz zu FileMaker API für PHP

Wenn FileMaker API für PHP bereits installiert ist, finden Sie Referenzinformationen in der Web-Server-Komponente Ihres FileMaker-Server-Einsatzes.

- Für IIS (Windows):
[Laufwerk]:\Programme\FileMaker\FileMaker Server\Documentation\PHP API Documentation\index.html
wobei [Laufwerk] sich auf das Laufwerk bezieht, auf dem die Web-Server-Komponente Ihres FileMaker-Server-Einsatzes gespeichert ist.
- Für Apache (macOS): /Library/FileMaker Server/Documentation/PHP API Documentation/index.html

FileMaker API für PHP-Support

Sie finden zusätzliche Informationen über die FileMaker API für PHP auf der Seite [FileMaker Support](#).

Verwendung der Klasse FileMaker

Um die Klasse FileMaker in Ihrer PHP-Lösung verwenden zu können, fügen Sie folgende Anweisung in Ihren PHP-Code ein:

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

FileMaker, Objekte der Klasse

Die Klasse FileMaker definiert Klassenobjekte, mit denen Sie Daten aus FileMaker Pro Advanced-Datenbanken abrufen können.

Klassenobjekt	Verwenden Sie das Objekt für
FileMaker-Datenbank	die Definition von Datenbankeigenschaften die Verbindung mit einer FileMaker Pro Advanced-Datenbank den Abruf von Informationen über FileMaker API für PHP
Befehl	die Erstellung von Befehlen, die Datensätze hinzufügen, Datensätze löschen, Datensätze bearbeiten, Suchabfragen durchführen und Scripts ausführen
Layout	die Arbeit mit Datenbanklayouts
Datensatz	die Arbeit mit Datensätzen
Feld	die Arbeit mit Felddaten
Bezugsmenge	die Arbeit mit Ausschnitt Datensätzen
Ergebnis	die Bearbeitung der Datensätze, die von Suchabfragen zurückgegeben werden
Fehler	die Überprüfung, ob ein Fehler aufgetreten ist die Bearbeitung von Fehlern

FileMaker-Befehlsobjekte

Die Klasse FileMaker definierte ein Basisbefehlsobjekt, mit dem ein bestimmter Befehl instantiiert wird und die Parameter des Befehls festgelegt werden. Zur Ausführung des Befehls muss die Methode `execute()` aufgerufen werden.

Die Klasse FileMaker definiert folgende bestimmte Befehle:

- Add, Befehl
- Compound Find, Befehl
- Delete, Befehl
- Duplicate, Befehl
- Edit, Befehl
- Find, Find All, Find Any
- Find Request, dieser Befehl wird dem Befehl „Compound Find Set“ hinzugefügt
- Script ausführen

Wichtig Befehle verfügen über unterschiedliche Rückgabewerte, wie durch die `FileMaker.php`-Klasse definiert. Beispielsweise geben einige Befehle den booleschen Wert `TRUE` oder ein `FileMaker_Error`-Objekt zurück. Andere Befehle liefern ein `FileMaker_Result`-Objekt, das eine komplette „Ergebnismenge“ von Datensätzen in einem Layout enthalten könnte. Um Speicherüberlaufprobleme zu vermeiden, berücksichtigen Sie die erwarteten Rückgabewerte der verwendeten Befehle. In der „Referenz zu FileMaker API für PHP“ auf Seite 72 finden Sie ausführliche Informationen über die Rückgabewerte für jeden Befehl.

Die grundlegenden Aufgaben der meisten PHP-Anwendungen werden nachfolgend beschrieben:

- „Arbeiten mit Datensätzen“ auf Seite 75
- „Ausführen von FileMaker-Scripts“ auf Seite 77
- „Ausführen von Suchabfragen“ auf Seite 84

Entschlüsseln von Daten für die Verwendung im FileMaker API

Wenn Ihre PHP-Anwendung Daten von einer Website abrufen, können diese Daten URL-kodiert sein. FileMaker API für PHP erwartet die Daten in entschlüsselter Form, nicht als URL-kodierte Zeichenfolgen. Generell sollten Sie die Funktion `urldecode()` aufrufen, wenn Sie Daten in Ihrer PHP-Anwendung abrufen.

Beispiel:

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

Hinweis Vermeiden Sie Zeichenfolgen mit `&`-Zeichen für das FileMaker API für PHP. Verwenden Sie einen umgekehrten Schrägstrich als Escape-Zeichen vor Sonderzeichen in Zeichenfolgen, die an FileMaker API für PHP übergeben werden.

Herstellen einer Verbindung zu einer FileMaker-Datenbank

Die Klasse `FileMaker` definiert ein Datenbankobjekt, das instantiiert wird, um eine Verbindung mit einem Server oder einer Datenbank herzustellen. Definieren Sie die Objekteigenschaften mit dem Klassenkonstruktor oder durch Aufruf der Methode `setProperty()`.

Beispiele

Herstellung einer Verbindung zu einem Server zum Abruf einer Datenbankliste:

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

Herstellung einer Verbindung zu einer bestimmten Datenbank auf einem Server:

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

Die Benutzernamen- und Passwortheigenschaften legen die Berechtigung für diese Verbindung fest.

Hinweis Die Eigenschaft `hostspec` wird automatisch auf den Wert `http://localhost` gesetzt. Da die PHP-Engine auf demselben Rechner wie die Web-Server-Komponente des Mastercomputers läuft, muss die Eigenschaft „hostspec“ nicht angegeben werden.

Arbeiten mit Datensätzen

Die Klasse `FileMaker` definiert ein Datensatzobjekt, das instantiiert wird, um mit Datensätzen zu arbeiten. Die Instanz eines Datensatzobjekts steht für einen Datensatz einer FileMaker Pro Advanced-Datenbank. Verwenden Sie ein Datensatzobjekt mit den Befehlen „Add“, „Delete“, „Duplicate“ und „Edit“, um die Daten im Datensatz zu ändern. Die Suchbefehle – „Find“, „Find All“, „Find Any“ und „Compound Find“ – geben ein Array der Datensatzobjekte zurück.

Erstellung eines Datensatzes

Datensätze können auf zwei unterschiedliche Arten erstellt werden:

- Verwenden Sie die Methode `createRecord()` und geben Sie dabei den Layoutnamen und optional ein Array an Feldwerten an. Die Werte können im neuen Datensatzobjekt auch einzeln angegeben werden.

Die Methode `createRecord()` speichert den neuen Datensatz nicht in der Datenbank. Rufen Sie die Methode `commit()` auf, um den Datensatz in der Datenbank zu speichern.

Beispiel:

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

Mit der Methode `commit()` von `FileMaker_Record` wird der Variablen `$result` der boolesche Wert `TRUE` zugewiesen und der neue Datensatz wird in der FileMaker-Datenbank angelegt, sofern keine Fehler auftreten.

Falls ein Fehler auftritt, enthält die Variable `$result` ein `FileMaker_Error`-Objekt. Prüfen Sie nach der Ausführung der Methode `commit()` auf Fehler.

- Verwenden Sie den Befehl „Add“: Verwenden Sie die Methode `newAddCommand()`, um ein `FileMaker_Command_Add`-Objekt zu erzeugen, das den Layoutnamen und das Array mit den Datensatzdaten enthält. Rufen Sie die Methode `execute()` auf, um den Datensatz in der Datenbank zu speichern.

Beispiel:

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

Mit der Methode `execute()` von `FileMaker_Command` enthält die Variable `$result` ein `FileMaker_Result`-Objekt mit sämtlichen Informationen über den erstellten Datensatz, sofern keine Fehler vorliegen.

Falls ein Fehler auftritt, enthält die Variable `$result` ein `FileMaker_Error`-Objekt. Prüfen Sie nach der Ausführung der Methode `execute()` auf Fehler.

Duplizieren von Datensätzen

Duplizieren Sie bestehende Datensätze mit dem Befehl „Duplicate“. Verwenden Sie die Methode `newDuplicateCommand()`, um ein `FileMaker_Command_Duplicate`-Objekt zu erstellen, das den Layoutnamen und die DatensatzID des Datensatzes enthält, den Sie duplizieren möchten. Duplizieren Sie den Datensatz anschließend durch Aufruf der Methode `execute()`.

Beispiel:

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

Bearbeiten von Datensätzen

Datensätze können auf zwei unterschiedliche Arten bearbeitet werden:

- Über den Befehl „Edit“. Verwenden Sie die Methode `newEditCommand()`, um ein `FileMaker_Command_Edit`-Objekt zu erstellen, das den Layoutnamen, die DatensatzID des Datensatzes, den Sie bearbeiten möchten, sowie das Array an Werten enthält, die Sie aktualisieren möchten. Bearbeiten Sie den Datensatz anschließend durch Aufruf der Methode `execute()`.

Beispiel:

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

- Mit einem Datensatzobjekt. Rufen Sie einen Datensatz aus der Datenbank ab, ändern Sie die Feldwerte und bearbeiten Sie den Datensatz durch Aufruf der Methode `commit()`.

Beispiel:

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

Löschen von Datensätzen

Datensätze können auf zwei unterschiedliche Arten gelöscht werden:

- Rufen Sie einen Datensatz aus der Datenbank ab und rufen Sie anschließend die Methode `delete()` auf.

Beispiel:

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

- Löschen Sie bestehende Datensätze mit dem Befehl „Delete“. Verwenden Sie die Methode `newDeleteCommand()`, um ein `FileMaker_Command_Delete`-Objekt zu erstellen, das den Layoutnamen und die DatensatzID des Datensatzes enthält, den Sie löschen möchten. Löschen Sie den Datensatz anschließend durch Aufruf der Methode `execute()`.

Beispiel:

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

Ausführen von FileMaker-Scripts

Bei einem FileMaker-Script handelt es sich um einen benannten Satz an Schrittschritten. Die Klasse `FileMaker` definiert mehrere Methoden, die es Ihnen ermöglichen, mit FileMaker-Scripts zu arbeiten, die in einer FileMaker Pro Advanced-Datenbank definiert sind. Weitere Informationen zu Web-kompatiblen Schrittschritten (Schrittschritte, die in einer Weblösung ausgeführt werden können) erhalten Sie unter „FileMaker-Scripts und Custom Web Publishing“ auf Seite 18.

Abruf der Liste verfügbarer Scripts

Verwenden Sie die Methode `listScripts()`, um eine Liste der verfügbaren Scripts aus der derzeit verbundenen Datenbank abzurufen. Die Methode `listScripts()` gibt ein Array der Scripts zurück, die über den Benutzernamen und das Passwort ausgeführt werden können, die bei der Definition der Datenbankverbindung angegeben wurden. (Weitere Informationen hierzu finden Sie unter „Herstellen einer Verbindung zu einer FileMaker-Datenbank“ auf Seite 75.)

Beispiel:

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

Ausführen von FileMaker-Scripts

Verwenden Sie die Methode `newPerformScriptCommand()`, um ein `FileMaker_Command_PerformScript`-Objekt zu erstellen, das das Layout, den Scriptnamen und Scriptparameter (falls vorhanden) enthält. Führen Sie das Script anschließend durch Aufruf der Methode `execute()` aus.

Wichtig Beim Ausführen eines FileMaker-Scripts hängt die Größe des zurückgegebenen `FileMaker_Result`-Objekts vom Verhalten des FileMaker-Scripts ab. Beispiel: Wenn ein FileMaker-Script in ein angegebenes Layout wechselt, können alle Datensätze aus der Tabelle dieses Layouts in der Ergebnismenge sein, und alle Datensätze in dieser Ergebnismenge könnten im `FileMaker_Result`-Objekt zurückgegeben werden. Um Speicherüberlaufprobleme zu vermeiden, berücksichtigen Sie die Daten, die ein FileMaker-Script zurückgeben kann, bevor Sie dieses FileMaker-Script in einer PHP-Anwendung ausführen.

Beispiel:

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

Ausführung eines Scripts vor der Ausführung eines Befehls

Verwenden Sie die Methode `setPreCommandScript()`, um ein Script anzugeben, das vor der Ausführung eines Befehls ausgeführt wird. Im folgenden Beispiel wird ein Find-Befehl verwendet, die Methode `setPreCommandScript()` kann jedoch für jeden Befehl eingesetzt werden.

Beispiel:

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

Ausführung eines Scripts vor der Sortierung einer Ergebnismenge

Verwenden Sie die Methode `setPreSortScript()`, um ein Script anzugeben, das nach der Erstellung einer Suchergebnismenge und vor der Sortierung dieser Menge ausgeführt wird. Siehe „Der Befehl „Find““ auf Seite 85.

Beispiel:

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

Ausführung eines Scripts nach der Erstellung einer Ergebnismenge

Verwenden Sie die Methode `setScript()`, um ein Script anzugeben, das nach der Erstellung einer Suchergebnismenge ausgeführt wird. Siehe „Der Befehl „Find““ auf Seite 85.

Beispiel:

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

Ausführungsreihenfolge von Scripts

Sie können die Methoden `setPreCommandScript()`, `setPreSortScript()` und `setScript()` in Verbindung mit den Methoden `setResultLayout()` und `addSortRule()` für einen einzelnen Befehl angeben.

FileMaker Server und die Web Publishing Engine verarbeiten diese Methoden in dieser Reihenfolge:

1. Ausführung des Scripts, das in der Methode `setPreCommandScript()` angegeben ist (falls angegeben)
2. Verarbeitung des Befehls, beispielsweise des Befehls „Find or Delete Record“
3. Ausführung des Scripts, das in der Methode `setPreSortScript()` angegeben ist (falls angegeben)
4. Sortierung der Suchergebnismenge, wenn die Methode `addSortRule()` angegeben wurde
5. Verarbeitung der Methode `setResultLayout()`, um das Layout zu wechseln (falls angegeben)
6. Ausführung des Scripts, das in der Methode `setScript()` angegeben ist (falls angegeben)
7. Rückgabe der endgültigen Suchergebnismenge

Wenn einer der obigen Schritte einen Fehlercode erzeugt, stoppt die Abarbeitung der Befehle und etwaige nachfolgende Schritte werden nicht ausgeführt. Jedoch bleiben alle vorangehenden Schritte in der Abfrage ausgeführt.

Betrachten Sie z. B. einen Befehl, der den aktuellen Datensatz löscht, die Datensätze sortiert und dann ein Script ausführt. Wenn die Methode `addSortRule()` ein nicht vorhandenes Feld angibt, löscht die Abfrage den aktuellen Datensatz und gibt dann Fehlercode 102 („Feld fehlt“) zurück, führt aber das Script nicht aus.

Das für die Methode `newFindCommand()` angegebene Layout wird verwendet, wenn die Suchabfrage bearbeitet wird. Wenn die Methode `setResultLayout()` zu einem anderen Layout umschaltet, ist das Fehlerobjekt für die Suchabfrage basierend auf dem Originallayout nicht mehr verfügbar. Um das Fehlerobjekt aus der Suchabfrage basierend auf dem Originallayout zu testen, prüfen Sie das Fehlerobjekt vor dem Umschalten des Layouts.

Beispiel:

```

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

// Suche durchführen
$result = $request->execute();

if (FileMaker::isError($result)) {
    if ($result->code = 401) {
        $findError = 'There are no Records that match that request: ' . ' (' .
            $result->code . ')';
    } else {
        $findError = 'Find Error: ' . $result->getMessage() . ' (' . $result->code
            . ')';
    }
}
}
$request->setResultLayout('Teachers');
// Zum Ergebnislayout umschalten
$result = $request->execute();

```

Arbeiten mit FileMaker-Layouts

Ein Layout ist eine Anordnung von Feldern, Objekten, Bildern und Layoutbereichen, die festlegen, wie Informationen beim Blättern, in der Seitenansicht oder beim Drucken angeordnet und angezeigt werden. Die Klasse FileMaker definiert mehrere Methoden, die es Ihnen ermöglichen, mit Layouts zu arbeiten, die in einer FileMaker Pro Advanced-Datenbank definiert sind. Informationen über Layouts können über unterschiedliche Objekte der Klasse FileMaker abgerufen werden.

Mit diesem Klassenobjekt	Verwenden Sie diese Methoden
Datenbank	<ul style="list-style-type: none"> ▪ <code>listLayouts()</code> ruft eine Liste der verfügbaren Layoutnamen ab. ▪ <code>getLayout()</code> ruft ein Layoutobjekt über den Layoutnamen ab.
Layout	<ul style="list-style-type: none"> ▪ <code>getName()</code> ruft den Layoutnamen eines bestimmten Layoutobjekts ab. ▪ <code>listFields()</code> ruft ein Array aller in einem Layout verwendeten Feldnamen ab. ▪ <code>getFields()</code> ruft ein assoziatives Array mit den Namen aller Felder als Schlüssel sowie die zugehörigen FileMaker_Field-Objekte als Array-Werte ab. ▪ <code>listValueLists()</code> ruft ein Array der Namen der Wertelisten ab. ▪ <code>listRelatedSets()</code> ruft ein Array der Namen der Bezugsmengen ab. ▪ <code>getDatabase()</code> gibt den Namen der Datenbank zurück.
Datensatz	<ul style="list-style-type: none"> ▪ <code>getLayout()</code> gibt das Layoutobjekt zurück, das mit einem bestimmten Datensatz verbunden ist.
Feld	<ul style="list-style-type: none"> ▪ <code>getLayout()</code> gibt das Layoutobjekt mit einem bestimmten Feld zurück.
Befehl	<ul style="list-style-type: none"> ▪ <code>setResultLayout()</code> gibt die Ergebnisse des Befehls in einem Layout zurück, das nicht mit dem aktuellen Layout identisch ist.

Verwenden von Ausschnitten

Ein Ausschnitt ist eine Tabelle, die Datenzeilen aus einem oder mehreren Bezugsdatensätzen enthält. Die Klasse FileMaker definiert ein Bezugsmengenobjekt und mehrere Methoden, die es Ihnen ermöglichen, mit Ausschnitten zu arbeiten, die in einer FileMaker Pro Advanced-Datenbank definiert sind.

Ein Bezugsmengenobjekt ist ein Array von Bezugsobjekten aus dem Bezugsausschnitt. Jedes Datensatzobjekt entspricht einer Datenzeile im Ausschnitt.

Auflistung der in einem bestimmten Layout definierten Ausschnitte

Verwenden Sie für ein bestimmtes Layoutobjekt die Methode `listRelatedSets()`, um eine Liste der Tabellennamen aller in diesem Layout definierten Ausschnitte abzurufen.

Beispiel:

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

Abrufen von Ausschnittnamen für ein bestimmtes Ergebnisobjekt

Verwenden Sie für ein bestimmtes FileMaker_Result-Objekt die Methode `getRelatedSets()`, um die Namen aller Ausschnitte in diesem Datensatz abzurufen.

Beispiel:

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

Abruf von Informationen über Ausschnitte für ein bestimmtes Layout

Verwenden Sie für ein bestimmtes Layoutobjekt die Methode `getRelatedSets()`, um ein Array an FileMaker_RelatedSet-Objekten abzurufen, die die Ausschnitte in dem Layout beschreiben. Das zurückgegebene Array ist ein assoziatives Array mit den Tabellennamen als Array-Schlüssel und den zugeordneten FileMaker_RelatedSet-Objekten als Array-Werte.

Beispiel:

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

Abrufen von Informationen für einen bestimmten Ausschnitt

Verwenden Sie für ein bestimmtes Layout-Objekt die Methode `getRelatedSet()`, um das FileMaker_RelatedSet-Objekt abzurufen, das einen bestimmten Ausschnitt beschreibt.

Beispiel:

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

Abrufen des Tabellennamens für einen Ausschnitt

Verwenden Sie die Methode `getName()`, um den Tabellennamen für diesen Ausschnitt abzurufen.

Beispiel:

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

Abrufen der Ausschnittdatensätze für einen bestimmten Datensatz

Verwenden Sie für ein bestimmtes Datensatzobjekt die Methode `getRelatedSet()`, um ein Array an Bezugsdatensätzen für einen bestimmten Ausschnitt des Datensatzes abzurufen.

Beispiel:

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

Erstellen eines neuen Datensatzes in einem Ausschnitt

Verwenden Sie die Methode `newRelatedRecord()`, um einen neuen Datensatz in der angegebenen Bezugsmenge zu erstellen und die Änderungen in der Datenbank durch Aufruf der Methode `commit()` zu bestätigen.

Beispiel:

```
//Neue Ausschnittzeile im Ausschnitt 'customer' erstellen
$new_row = $currentRecord->newRelatedRecord('customer');

//Feldwerte in der neuen Ausschnittzeile festlegen
$new_row->setField('customer::name', $newName);
$new_row->setField('customer::company', $newCompany);

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

Löschen eines Datensatzes aus einem Ausschnitt

Verwenden Sie die Methode `delete()`, um einen Datensatz aus einem Ausschnitt zu löschen.

Beispiel:

```
$relatedSet = $currentRecord->getRelatedSet('customers');
/* Runs through each of the portal rows */
foreach ($relatedSet as $nextRow) {
    $nameField = $nextRow->getField('customer::name')
    if ($nameField == $badName) {
        $result = $newRow->delete();
    }
}
```

Verwenden von Wertelisten

Eine Werteliste ist eine vordefinierte Auswahl. Die Klasse FileMaker definiert mehrere Methoden, die es Ihnen ermöglichen, mit Wertelisten zu arbeiten, die in einer FileMaker Pro Advanced-Datenbank definiert sind.

Abrufen der Namen aller Wertelisten für ein bestimmtes Layout

Verwenden Sie für ein bestimmtes Layoutobjekt die Methode `listValueLists()`, um ein Array abzurufen, das die Namen der Wertelisten enthält.

Beispiel:

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

Abrufen eines Arrays aller Wertelisten für ein bestimmtes Layout

Verwenden Sie für ein bestimmtes Layoutobjekt die Methode `getValueListsTwoFields()`, um ein Array abzurufen, das die Werte aller Wertelisten enthält. Bei dem zurückgegebenen Array handelt es sich um ein assoziatives Array. Die Array-Schlüssel sind die Namen der Wertelisten, und Array-Werte sind assoziative Arrays, die die Anzeigenamen und ihre entsprechenden Auswahlmöglichkeiten jeder Werteliste auflisten.

Beispiel:

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

Hinweis Obwohl die Methode `getValueLists()` noch im FileMaker API für PHP unterstützt wird, wird sie abgelehnt. Verwenden Sie stattdessen die Methode `getValueListsTwoFields()`.

Beziehen von Werten für eine benannte Werteliste

Verwenden Sie für ein bestimmtes Layoutobjekt die Methode `getValueList()`, um ein Array an Auswahlmöglichkeiten abzurufen, die für die benannte Werteliste definiert sind. Das resultierende Array ist ein assoziatives Array, das die angezeigten Werte aus dem zweiten Feld der Werteliste als Schlüssel und die zugehörigen gespeicherten Werte aus dem ersten Feld als Array-Werte enthält.

Abhängig von den ausgewählten Optionen im Dialogfeld „Felder für Werteliste angeben“ in der FileMaker-Datenbank gibt die Methode `getValueListTwoFields()` nur den Wert im ersten Feld, nur den Wert im zweiten Feld oder den Wert in beiden Feldern einer Werteliste als gespeicherte und angezeigte Werte zurück.

- Wenn **Auch Werte aus zweitem Feld anzeigen** nicht ausgewählt ist, gibt die Methode `getValueListTwoFields()` den Wert aus dem ersten Feld der Werteliste sowohl als gespeicherten Wert als auch als angezeigten Wert zurück.
- Wenn sowohl **Auch Werte aus zweitem Feld anzeigen** als auch **Werte nur aus dem zweiten Feld anzeigen** ausgewählt ist, gibt die Methode `getValueListTwoFields()` den Wert aus dem ersten Feld als gespeicherten Wert und den Wert aus dem zweiten Wert als angezeigten Wert zurück.

- Wenn **Auch Werte aus zweitem Feld anzeigen** ausgewählt und **Werte nur aus dem zweiten Feld anzeigen** nicht ausgewählt ist, gibt die Methode `getValueListTwoFields()` den Wert aus dem ersten Feld als gespeicherten Wert und beide Werte aus dem ersten und zweiten Wert als angezeigten Wert zurück.

Verwenden Sie einen Iterator mit der Methode `getValueListTwoFields()`, um den angezeigten und gespeicherten Wert zu finden.

Beispiel:

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

Hinweise

- Obwohl die Methode `getValueList()` noch im FileMaker API für PHP unterstützt wird, wird sie abgelehnt. Verwenden Sie stattdessen die Methode `getValueListTwoFields()`.
- Stellen Sie bei Verwendung der Methode `getValueListTwoFields()` sicher, dass Sie eine `foreach`-Schleife für das zugehörige Array verwenden. Verwenden Sie keine `for`-Schleife, da dies zu unerwarteten Ergebnissen führen kann.

Ausführen von Suchabfragen

Die Klasse FileMaker definiert vier Arten von Suchbefehlsobjekten:

- Befehl „Find All“ Siehe „Der Befehl „Find All““ auf Seite 85.
- Befehl „Find Any“ Siehe „Der Befehl „Find Any““ auf Seite 85.
- Befehl „Find“ Siehe „Der Befehl „Find““ auf Seite 85.
- Befehl „Compound Find“ Siehe „Der Befehl „Compound Find““ auf Seite 86.

Zudem definiert die Klasse FileMaker mehrere Methoden, die für alle vier Suchbefehlstypen verwendet werden können:

- Verwenden Sie die Methode `addSortRule()`, um eine Regel hinzuzufügen, die festlegt, wie die Ergebnismenge sortiert wird. Verwenden Sie die Methode `clearSortRules()`, um alle vorhandenen Sortierregeln zu löschen.
- Verwenden Sie die Methode `setLogicalOperator()`, um zwischen logischen UND- und ODER-Suchen zu wechseln.
- Verwenden Sie die Methode `setRange()`, um nur einen Teil der Ergebnismenge anzufordern. Verwenden Sie die Methode `getRange()`, um die aktuelle Bereichsdefinition abzurufen.

Mit der Methode `setRange()` können Sie die Leistung Ihrer Lösung verbessern, da weniger Datensätze von der Suchabfrage zurückgegeben werden. Wenn eine Suchabfrage beispielsweise 100 Datensätze zurückgibt, können Sie die Ergebnismenge in fünf Gruppen mit je 20 Datensätzen aufteilen, anstatt alle 100 Datensätze gleichzeitig zu verarbeiten.

- FileMaker-Scripts können zusammen mit Suchbefehlen ausgeführt werden.
 - Um ein Script vor der Ausführung eines Suchbefehls auszuführen, verwenden Sie die Methode `setPreCommandScript()`.
 - Um ein Script vor der Sortierung der Ergebnismenge auszuführen, verwenden Sie die Methode `setPreSortScript()`.
 - Um ein Script nach der Erstellung und vor der Sortierung einer Ergebnismenge auszuführen, verwenden Sie die Methode `setScript()`.

Der Befehl „Find All“

Verwenden Sie den Befehl „Find All“, um alle Datensätze aus einem angegebenen Layout abzurufen. Verwenden Sie die Methode `newFindAllCommand()` und geben Sie dabei ein bestimmtes Layout an, um ein `FileMaker_Command_FindAll`-Objekt zu erstellen. Führen Sie die Suchabfrage anschließend durch Aufruf der Methode `execute()` aus.

Beispiel:

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

Hinweis Wenn Sie den Befehl „Find All“ verwenden, vermeiden Sie Speicherüberlaufprobleme, indem Sie eine Standard-Maximalzahl von Datensätzen pro Seite angeben.

Der Befehl „Find Any“

Verwenden Sie den Befehl „Find Any“, um einen zufällig ermittelten Datensatz aus einem angegebenen Layout abzurufen. Verwenden Sie die Methode `newFindAnyCommand()` und geben Sie dabei ein bestimmtes Layout an, um ein `FileMaker_Command_FindAny`-Objekt zu erstellen. Führen Sie die Suchabfrage anschließend durch Aufruf der Methode `execute()` aus.

Beispiel:

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

Der Befehl „Find“

Verwenden Sie die Methode `newFindCommand()` und geben Sie dabei ein bestimmtes Layout an, um ein `FileMaker_Command_Find`-Objekt zu erstellen. Führen Sie die Suchabfrage anschließend durch Aufruf der Methode `execute()` aus.

Hinweis Stellen Sie sicher, dass der Layoutname eindeutig ist. Wenn Ihre Datenbank zwei Layouts mit dem gleichen Namen enthält, kann das FileMaker API für PHP nicht zwischen ihnen unterscheiden. Zusätzlich unterscheidet das API nicht zwischen Groß- und Kleinschreibung. Wenn Ihre Datenbank zum Beispiel ein Layout namens „Websites“ und ein anderes Layout namens „WebSites“ hat, kann das API nicht zwischen ihnen unterscheiden.

Um der Suchabfrage Kriterien hinzuzufügen, verwenden Sie die Methode `addFindCriterion()`. Verwenden Sie die Methode `clearFindCriteria()`, um alle vorhandenen Suchkriterien zu löschen.

Beispiele

So suchen Sie einen Datensatz nach Feldname:

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

Hinzufügen einer Sortierfolge:

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

Der Befehl „Compound Find“

Mit dem Befehl „Compound Find“ können Sie mehrere Suchabfrageobjekte zu einem Befehl zusammenfassen. Es gibt mehrere Möglichkeiten einen CompoundFind-Befehl zu erstellen:

- Erstellen Sie durch Aufruf der Methode `newCompoundFindCommand()` ein `FileMaker_Command_CompoundFind`-Objekt.
- Erstellen Sie durch Aufruf der Methode `newFindRequest()` eines oder mehrere `FileMaker_Command_FindRequest`-Objekte.
- Verwenden Sie die Methode `add()`, um dem Befehlsobjekt „Compound Find“ die Suchabfrageobjekte hinzuzufügen.
- Führen Sie den Befehl „Compound Find“ durch Aufruf der Methode `execute()` aus.

Beispiel:**Befehl „Compound Find“**

```
// Das Befehlsobjekt „Compound Find“ erstellen
$compoundFind = $fm->newCompoundFindCommand('Form View');

// Erste Suchabfrage erstellen
$findreq1 = $fm->newFindRequest('Form View');

// Zweite Suchabfrage erstellen
$findreq2 = $fm->newFindRequest('Form View');

// Dritte Suchabfrage erstellen
$findreq3 = $fm->newFindRequest('Form View');

// Suchkriterium für erste Suchabfrage angeben
$findreq1->addFindCriterion('Quantity in Stock', '<100');

// Suchkriterium für zweite Suchabfrage angeben
$findreq2->addFindCriterion('Quantity in Stock', '0');

// Suchkriterium für dritte Suchabfrage angeben
$findreq3->addFindCriterion('Cover Photo Credit', 'The London Morning
News');

// Suchabfragen dem zusammengesetzten Suchbefehl hinzufügen
$compoundFind->add(1,$findreq1);
$compoundFind->add(2,$findreq2);
$compoundFind->add(3,$findreq3);

// Sortierfolge festlegen
$compoundFind->addSortRule('Title', 1, FILEMAKER_SORT_DESCEND);

// Zusammengesetzten Suchbefehl ausführen
$result = $compoundFind->execute();

// Datensätze aus Ergebnismenge abrufen
$records = $result->getRecords();

// Anzahl an gefundenen Datensätzen drucken
echo 'Found '. count($records) . " results.<br><br>";
```

Verarbeitung der Datensätze in einer Ergebnismenge

- Rufen Sie durch Aufruf der Methode `getRecords()` ein Array auf, das alle Datensätze in der Ergebnismenge enthält. Jeder Eintrag des Arrays ist ein FileMaker-Record-Objekt oder eine Instanz des im API für die Instantiierung von Datensätzen festgelegten Klassennamens. Das Array kann leer sein, wenn die Ergebnismenge keine Datensätze enthält.
- Rufen Sie durch Aufruf der Methode `getFields()` eine Liste der Feldnamen aller Felder in der Ergebnismenge ab. Die Methode gibt nur die Feldnamen zurück. Wenn Sie weitere Informationen zu den Feldern benötigen, verwenden Sie das entsprechende Layoutobjekt.
- Rufen Sie die Anzahl an Datensätzen in der gesamten Ergebnismenge durch Aufruf der Methode `getFoundSetCount()` ab.
- Rufen Sie die Anzahl an Datensätzen in der gefilterten Ergebnismenge durch Aufruf der Methode `getFetchCount()` ab. Wenn keine Bereichsparameter für den Suchbefehl angegeben wurden, ist dieser Wert mit dem Ergebnis der Methode `getFoundSetCount()` identisch. Er ist immer mit dem Wert von `count($response->getRecords())` identisch.
- Verwenden Sie für einen bestimmten Datensatz die Methode `getField()`, um den Inhalt eines Felds als Zeichenfolge zurückzugeben.
- Verwenden Sie für einen bestimmten Datensatz die Methode `getFieldAsTimestamp()`, um den Inhalt eines Felds als Unix-Zeitstempel (die PHP-interne Darstellung eines Datums) zurückzugeben.
 - Wenn es sich bei dem Feld um ein Datumsfeld handelt, bezieht sich der Zeitstempel auf das Felddatum um Mitternacht.
 - Wenn es sich bei dem Feld um ein Zeitfeld handelt, bezieht sich der Zeitstempel auf die Uhrzeit am 1. Januar 1970.
 - Wenn es sich bei dem Feld um ein Zeitstempelfeld handelt, verweist der FileMaker-Zeitstempelwert direkt auf den Unix-Zeitstempel.
 - Wenn es sich bei dem angegebenen Feld weder um ein Datums- noch um ein Zeitfeld handelt oder wenn sich der erstellte Zeitstempel außerhalb des gültigen Bereichs befände, gibt die Methode `getFieldAsTimestamp()` ein `FileMaker_Error`-Objekt zurück.
- Verwenden Sie für einen bestimmten Datensatz die Methode `getContainerData()`, um den Inhalt eines Containerfeldobjekts als Binärdaten zurückzugeben.

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

```
echo $fm->getContainerData($_GET['-url']);
```

- Verwenden Sie für einen bestimmten Datensatz die Methode `getContainerDataURL()`, um eine vollständig qualifizierte URL für das Containerfeldobjekt zurückzugeben:

```
// Verwenden Sie für Bilder den HTML-Tag img
```

```
echo '';
```

```
// Verwenden Sie für Filme und PDF-Dateien den HTML-Tag embed
```

```
//echo '<embed src="'. $fm->
```

```
getContainerDataURL($record->getField('container')) .'">';
```

Begrenzen der Ausschnittzeilen, die von Suchabfragen zurückgegeben werden

In einer Lösung mit vielen Datensätzen kann das Abfragen und Sortieren von Datensätzen einige Zeit in Anspruch nehmen. Verwenden Sie die Methode `setRelatedSetsFilters()` mit Suchabfragen, um die Anzahl an Datensätzen einzuschränken, die in einer Bezugsmenge angezeigt werden. Die Methode `setRelatedSetsFilters()` akzeptiert zwei Argumente:

- einen Bezugsmengen-Filterwert: `layout` oder `none`.
 - Wenn Sie den Wert `none` angeben, gibt die Web Publishing Engine alle Zeilen im Ausschnitt zurück und die Ausschnittdatensätze werden nicht vorsortiert.
 - Wenn Sie den Wert `layout` angeben, werden die im FileMaker Pro Advanced-Dialogfeld „Ausschnitteinstellungen“ angegebenen Einstellungen respektiert. Die Datensätze werden basierend auf der im Dialogfeld „Ausschnitteinstellungen“ definierten Sortierung sortiert, wobei die Datensatzmenge gefiltert ist, um mit der angegebenen ersten Zeile zu beginnen.
- die minimale Anzahl an zurückgegebenen Ausschnittdatensätzen: eine Ganzzahl oder `all`.
 - Dieser Wert wird nur verwendet, wenn die Einstellung **Vertikaler Rollbalken** im Dialogfeld „Ausschnitteinstellungen“ aktiviert ist. Wenn Sie eine Ganzzahl angeben, wird diese Anzahl an Zeilen nach der ersten Zeile zurückgegeben. Wenn Sie `all` angeben, gibt die Web Publishing Engine alle Bezugsdatensätze zurück.
 - Wenn die Einstellung **Vertikaler Rollbalken** deaktiviert ist, bestimmt die Einstellung **Anzahl an Zeilen** im Dialogfeld „Ausschnitteinstellungen“ die maximale Anzahl an zurückgegebenen Bezugsdatensätzen.

Hinweis Die Einstellung **Ausschnittdatensätze filtern** im Dialogfeld „Ausschnitteinstellungen“ wird für PHP-Abfragen nicht unterstützt. Etwaige Formeln, die für die Einstellung **Ausschnittdatensätze filtern** angegeben sind, werden ignoriert.

Vorabüberprüfung von Befehlen, Datensätzen und Feldern

Mit der Klasse FileMaker können Sie Felddaten in einer PHP-Lösung auf dem Web-Server *vorab überprüfen*, bevor die Daten in der Datenbank bestätigt werden.

Vorabüberprüfung ist standardmäßig deaktiviert. Aktivieren Sie die Vorabüberprüfung über die CLI. Siehe CLI-Hilfe.

Bevor Sie sich zu einer Vorabüberprüfung entschließen, berücksichtigen Sie, wie viele Datenwerte der Webbenutzer eingibt. Wenn der Benutzer nur wenige Felder aktualisiert, könnten Sie die Leistung verbessern, indem Sie die Vorabüberprüfung nicht verwenden. Wenn der Benutzer jedoch Daten für viele Felder eingibt, kann die Vorabüberprüfung den Benutzer vor der Abweisung von Datensätzen durch die Datenbank aufgrund von Überprüfungsfehlern bewahren.

Mit der Klasse FileMaker überprüft die PHP Engine folgende Feldbeschränkungen vorab:

- `not empty`
Gültige Daten sind nicht leere Zeichenfolgen. Die Daten müssen mindestens ein Zeichen enthalten.
- `numeric only`
Gültige Daten enthalten nur numerische Zeichen.
- `maximum number of characters`
Gültige Daten enthalten höchstens die angegebene Anzahl an Zeichen.

- Four-digit year

Gültige Daten sind eine Zeichenfolge, die ein Datum mit einer vierstelligen Jahreszahl im Format M/T/JJJJ wiedergibt, wobei M eine Zahl zwischen 1 und einschließlich 12, T eine Zahl zwischen 1 und einschließlich 31 und JJJJ eine vierstellige Zahl zwischen 0001 und einschließlich 4000 ist. 1/30/3030 ist beispielsweise ein gültiger Wert einer vierstelligen Jahreszahl. 4/31/2019 ist hingegen ein ungültiger Wert einer vierstelligen Jahreszahl, da der April nur 30 Tage hat. Die Datumsüberprüfung unterstützt Schrägstriche (/), umgekehrte Schrägstriche (\) und Bindestriche (-) als Trennzeichen. Die Zeichenfolge darf jedoch keine unterschiedlichen Trennzeichen enthalten. 1\30-2019 ist beispielsweise ungültig.

- time of day

Gültige Daten sind Zeichenfolgen, die einen 12-Stunden-Zeitwert in einem der folgenden Formate wiedergeben:

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

Dabei ist h eine Zahl zwischen 1 und einschließlich 12, m und s sind Zahlen zwischen 1 und einschließlich 60.

Die Vorabüberprüfung der PHP Engine unterstützt die implizite Überprüfung von Felddaten nach Feldtyp:

- date

Ein Feld, das als Datumsfeld definiert ist, wird nach den Regeln der Überprüfung für vierstellige Jahreszahlen überprüft, abgesehen davon, dass der Jahreswert aus 0-4 Stellen bestehen kann (der Jahreswert kann leer sein). 1/30 ist beispielsweise ein gültiges Datum, obwohl kein Jahr angegeben ist.

- time

Ein Feld, das als Zeitfeld definiert ist, wird nach den Regeln der Überprüfung für Tageszeiten überprüft, abgesehen davon, dass die Stundenkomponente (H) eine Zahl zwischen 1 und 24 sein kann, um 24-Stunden-Zeitwerte zu ermöglichen.

- timestamp

Ein Feld, das als Zeitstempelfeld definiert ist, wird für die Zeitkomponente nach den Regeln der Überprüfung für Zeiten und für die Datumskomponente nach den Regeln für die Überprüfung von Datumsangaben überprüft.

Die Klasse FileMaker kann nicht alle Feldüberprüfungsoptionen, die in FileMaker Pro Advanced verfügbar sind, vorab überprüfen. Folgende Überprüfungsoptionen können nicht vorab überprüft werden, da sie vom Status aller Daten in der Datenbank zum Zeitpunkt der Bestätigung der Daten abhängig sind.

- Eindeutiger Wert
- Vorhandener Wert
- Im Bereich
- Aus Werteliste
- Überprüfung durch Berechnung

Vorabüberprüfung von Datensätzen in einem Befehl

Verwenden Sie für ein Befehlsobjekt die Methode `validate()`, um ein Feld oder den gesamten Befehl nach den Regeln der Vorabüberprüfung, die über die PHP Engine durchgesetzt werden können, zu überprüfen. Wenn Sie das optionale Argument für Feldnamen übergeben, wird nur dieses Feld vorab überprüft.

Wenn die Vorabüberprüfung bestanden wird, gibt die Methode `validate()` `TRUE` zurück. Wenn die Vorabüberprüfung fehlschlägt, gibt die Methode `validate()` ein `FileMaker_Error_Validation` Vorabüberprüfung mit den Fehlerursachen zurück.

Vorabüberprüfung von Datensätzen

Verwenden Sie für ein Datensatzobjekt die Methode `validate()`, um ein Feld oder alle Felder des Datensatzes nach den Regeln der Vorabüberprüfung, die über die PHP Engine durchgesetzt werden können, zu überprüfen. Wenn Sie das optionale Argument für Feldnamen übergeben, wird nur dieses Feld vorab überprüft.

Wenn die Vorabüberprüfung bestanden wird, gibt die Methode `validate()` `TRUE` zurück. Wenn die Vorabüberprüfung fehlschlägt, gibt die Methode `validate()` ein `FileMaker_Error_Validation` Vorabüberprüfung mit den Fehlerursachen zurück.

Vorabüberprüfung von Feldern

Verwenden Sie für ein Feldobjekt die Methode `validate()`, um zu ermitteln, ob ein bestimmter Wert für ein Feld gültig ist.

Wenn die Vorabüberprüfung bestanden wird, gibt die Methode `validate()` `TRUE` zurück. Wenn die Vorabüberprüfung fehlschlägt, gibt die Methode `validate()` ein `FileMaker_Error_Validation` Vorabüberprüfung mit den Fehlerursachen zurück.

Bearbeitung der Überprüfungsfehler

Wenn die Vorabüberprüfung fehlschlägt, enthält das zurückgegebene `FileMaker_Error_Validation`-Objekt ein Array mit drei Elementen für jeden Überprüfungsfehler:

1. Das Feldobjekt, das die Vorabüberprüfung nicht bestanden hat
2. Eine Überprüfungskonstante, die die Überprüfungsregel enthält, die nicht bestanden wurde:
 - 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. Den Wert, der in das Feld eingegeben wurde, das die Vorabüberprüfung nicht bestanden hat

Sie können zudem folgende Methoden für FileMaker_Error_Validation-Objekte verwenden:

- Verwenden Sie die Methode `isValidatationError()`, um zu testen, ob es sich bei dem Fehler um einen Überprüfungsfehler handelt.
- Verwenden Sie die Methode `numErrors()`, um die Anzahl an nicht bestandenen Überprüfungsregeln abzurufen.

Beispiel:

```
//Eine Add-Abfrage erstellen
$addrequest = $fm->newAddCommand('test', array('join' => 'added', 'maxchars'
=> 'abcx', 'field' => 'something' , 'numericonly' => 'abc'));

//Alle Felder überprüfen
$result = $addrequest->validate();

//Wenn die Methode „validate()“ Fehler zurückgegeben hat, Namen des Felds,
Fehlernummer und den Wert, der die Überprüfung nicht bestanden hat, drucken
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";
    }
}
```

Ergebnis

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

Fehlerbehandlung

Die Klasse FileMaker definiert das FileMaker_Error-Objekt, um Sie bei der Behandlung von Fehlern in einer PHP-Lösung zu unterstützen.

Ein Fehler kann auftreten, wenn ein Befehl ausgeführt wird. Wenn ein Fehler auftritt, gibt der Befehl ein FileMaker_Error-Objekt zurück. Es empfiehlt sich, den Fehler zu überprüfen, der bei der Ausführung eines Befehls aufgetreten ist.

Verwenden Sie folgende Methoden, um genauere Informationen über den im FileMaker_Error-Objekt angegebenen Fehler zu erhalten.

- Testen Sie, ob es sich bei einer Variablen um ein FileMaker_Error-Objekt handelt, indem Sie die Methode `isError()` aufrufen.
- Rufen Sie die Fehleranzahl über die Methode `numErrors()` ab.
- Rufen Sie ein Array von Arrays, die die aufgetretenen Fehler beschreiben, über die Methode `getErrors()` ab.
- Zeigen Sie die Fehlermeldung über die Methode `getMessage()` an.

Beispiel:

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

Weitere Informationen über die mit dem FileMaker_Error-Objekt zurückgegebenen Fehlercodes erhalten Sie in Anhang A „Fehlercodes für Custom Web Publishing“.

Kapitel 9

Einstellen, Testen und Überwachen einer Site

Dieses Kapitel enthält Informationen über das Einstellen und Testen einer Custom Web Publishing-Site vor ihrem Einsatz in einer Produktionsumgebung. Zudem finden Sie hier Informationen zur Verwendung von Protokolldateien, um die Site während der Tests und nach der Freigabe zu überwachen.

Einstellen einer Custom Web Publishing-Site

Bevor Sie Ihre Site richtig testen können, müssen Sie die erforderlichen Dateien an die richtigen Speicherorte auf dem bzw. den Servern kopieren oder verschieben.

1. Folgen Sie den Anleitungen in Kapitel 2 „Vorbereiten von Datenbanken für Custom Web Publishing“.
2. Stellen Sie sicher, dass Custom Web Publishing mittels CLI aktiviert und richtig konfiguriert wurde. Weitere Informationen finden Sie in der [FileMaker Server Hilfe](#).
3. Stellen Sie sicher, dass der Web-Server und die Web Publishing Engine laufen.
4. Kopieren oder verschieben Sie Ihre Website-Dateien auf die Web-Server-Komponente Ihres FileMaker-Server-Einsatzes.

Kopieren oder verschieben Sie Ihre Website-Dateien in folgendes Verzeichnis:

- Für IIS (Windows) durch HTTP oder HTTPS:
`[Laufwerk]:\Programme\FileMaker\FileMaker Server\HTTPServer\Conf`
wobei sich [Laufwerk] auf das Laufwerk Ihres FileMaker Server-Einsatzes bezieht.
- Apache (macOS) durch HTTP:
`/Library/FileMaker Server/HTTPServer/htdocs`
- Apache (macOS) durch HTTPS:
`/Library/FileMaker Server/HTTPServer/htdocs/httpsRoot`

Hinweis Wenn Sie ein eigenes SSL-Zertifikat importieren, verwenden Datenbank-Server-Client-Verbindungen SSL und HTTP-Verbindungen werden über HTTPS geroutet. Verwenden Sie mit einem eigenen SSL-Zertifikat das HTTPS-Verzeichnis für die Bereitstellung von PHP-Site-Dateien.

5. Wenn nicht bereits geschehen, kopieren oder verschieben Sie referenzierte Containerfeldobjekte in das richtige Verzeichnis auf dem Mastercomputer.
 - Falls die Containerfelder die tatsächlichen Dateien in der FileMaker-Datenbank speichern, ist mit dem Inhalt der Containerfelder keine Aktion erforderlich, wenn die Datenbankdatei richtig auf der Datenbank-Server-Komponente des FileMaker Server-Einsatzes bereitgestellt und zugänglich ist.
 - Wenn ein Datenbank-Containerfeld einen Dateiverweis anstelle der eigentlichen Datei speichert, dann muss das referenzierte Containerobjekt im FileMaker Pro Advanced-Webordner gespeichert sein, wenn der Datensatz erstellt oder bearbeitet wird. Um Ihre Site einzustellen, kopieren oder verschieben Sie die referenzierten Container in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software.
 - Wenn Sie mithilfe von FileMaker Pro Advanced eine Datenbank mit Containerfeldern hochladen, die Objekte extern speichern, werden dabei die extern gespeicherten Containerfelddaten zu FileMaker Server hochgeladen. In der [FileMaker Pro Advanced Hilfe](#) finden Sie Informationen zum Übertragen der Datenbankdateien auf FileMaker Server.
 - Wenn Sie eine Datenbank, die ein Containerfeld mit extern gespeicherten Objekten verwendet, manuell hochladen, müssen Sie die referenzierten Objekte in einen Unterordner des Ordners RC_Data_FMS kopieren. Siehe dazu „Containerfelder mit extern gespeicherten Daten“ auf Seite 17.
6. Kopieren Sie etwaige zusätzliche Komponenten Ihrer Web-Anwendung auf den Mastercomputer. Für Custom Web Publishing mit XML verarbeitet Ihre Web-Anwendung die XML-Daten, bevor sie sie an eine andere Anwendung oder den Client sendet.

Testen einer Custom Web Publishing-Site

Bevor Sie Benutzern mitteilen, dass Ihre Custom Web Publishing-Site verfügbar ist, sollten Sie überprüfen, ob Aussehen und Funktionsweise Ihren Vorstellungen entsprechen.

- Testen Sie Funktionen wie das Suchen, Hinzufügen, Löschen und Sortieren von Datensätzen mit unterschiedlichen Konten und Zugriffsrechten.
- Überprüfen Sie, ob sich die Berechtigungen wie erwartet verhalten, indem Sie sich mit verschiedenen Konten anmelden. Stellen Sie sicher, dass unbefugte Benutzer nicht auf Ihre Daten zugreifen oder sie ändern können.
- Überprüfen Sie alle Scripts, um sicherzustellen, dass das Ergebnis wie erwartet ist (siehe „FileMaker-Scripts und Custom Web Publishing“ auf Seite 18 für Informationen zur Gestaltung Web-freundlicher Scripts).
- Testen Sie Ihre Site mit unterschiedlichen Betriebssystemen und Webbrowsern.
- Für die Erstellung von Lösungen, die FileMaker API für PHP verwenden, wird empfohlen, Ihre Lösungen mit aktivierter Cookie-Unterstützung zu erstellen. Das FileMaker API für PHP hat bessere Reaktionszeiten, wenn Cookies aktiviert sind. Cookies sind nicht erforderlich, um Custom Web Publishing-Funktionen zu nutzen, aber Cookies gestatten der Web Publishing Engine, Sitzungsinformationen im Cache abzulegen.

Hinweis Sie können Ihre Website auf dem Mastercomputer anzeigen und testen, ohne eine Netzwerkverbindung zu nutzen. Verwenden Sie `http://127.0.0.1/` in dem URL.

- Verwenden Sie für PHP-Lösungen `http://127.0.0.1/<site_path>`. Dabei ist `<site_path>` der relative Pfad zur Homepage Ihrer Site.
- Informationen zur URL-Syntax in XML-Lösungen finden Sie unter „Erläuterung der URL-Syntax für XML-Daten und Containerobjekte“ auf Seite 27.

Stylesheets zum Testen der XML-Ausgabe

Beispiele

Hier sind zwei Beispiele für XSLT-Stylesheets, die sich zum Testen von XML-Ausgabe eignen.

Das folgende Stylesheet-Beispiel gibt die angeforderten XML-Daten ohne Umwandlung aus. Dieses Stylesheet ist nützlich für die Anzeige der tatsächlichen XML-Daten, die die Web Publishing Engine verwendet.

```
<?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>
```

Bei der Fehlerbehebung in einem Stylesheet können Sie das folgende Beispiel des HTML-Tags `<textarea>` verwenden, um das XML-Quelldokument anzuzeigen, auf das über das Stylesheet in einem scrollbaren Textbereich zugegriffen wurde.

```
<?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>
```

Überwachen Ihrer Website

Sie können anhand der folgenden Typen von Protokolldateien Ihre Custom Web Publishing-Site überwachen und Informationen über Webbenutzer sammeln, die Ihre Site besuchen:

- Web-Server-Zugriff und Fehlerprotokolle
- Web Publishing Engine-Protokoll
- Web-Server-Modul-Fehlerprotokoll
- Tomcat-Protokolle

Verwenden der Web-Server-Zugriffs- und Fehlerprotokolle

IIS (Windows): Der Microsoft IIS-Web-Server generiert eine Zugriffsprotokolldatei und zeigt Fehler in der Windows-Ereignisanzeige, anstatt sie in eine Protokolldatei zu schreiben. Die Zugriffsprotokolldatei, standardmäßig im W3C Extended Log File Format, ist eine Aufzeichnung aller beim Web-Server eingehenden HTTP-Abfragen. Sie können auch das W3C Common Logfile Format für das Zugriffsprotokoll verwenden. Weitere Informationen finden Sie in der Dokumentation zum Microsoft IIS-Web-Server.

Apache (macOS): Der Apache-Web-Server generiert eine Zugriffsprotokolldatei und eine Fehlerprotokolldatei. Die Apache-Zugriffsprotokolldatei, standardmäßig im W3C Common Logfile Format, ist eine Aufzeichnung aller beim Web-Server eingehenden HTTP-Abfragen. Das Apache-Fehlerprotokoll ist eine Aufzeichnung von Problemen bei der Verarbeitung von HTTP-Abfragen. Weitere Informationen finden Sie in der Dokumentation zum Apache-Web-Server.

Hinweis Informationen über das W3C Common Logfile Format und das W3C Extended Log File Format finden Sie auf der Website des World Wide Web Consortium: www.w3.org.

Verwenden des Web Publishing Engine-Protokolls

Die Web Publishing Engine generiert eine Protokolldatei namens wpe.log, die Einträge zu aufgetretenen Web Publishing Engine-Fehlern einschließlich Anwendungs-, Nutzungs- und Systemfehlern enthält.

Die Datei wpe.log wird generiert, wenn die Web Publishing Engine läuft. (Siehe Admin Console, um zu prüfen, ob die Web Publishing Engine läuft.)

Die Datei wpe.log befindet sich im folgenden Verzeichnis in der Web Publishing Engine-Komponente Ihres FileMaker Server-Einsatzes:

- Windows:
[Laufwerk]:\Programme\FileMaker\FileMaker Server\Logs\wpe.log
wobei [Laufwerk] das Hauptlaufwerk ist, von dem das System startet.
- macOS: /Library/FileMaker Server/Logs/wpe.log

Web Publishing Engine-Protokollformat

Die Datei wpe.log file verwendet das folgende Format für die Einträge:

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

Dabei gilt:

- [TIMESTAMP_GMT] sind Datum und Zeit des Eintrags in Greenwich Mean Time (GMT).
- [WPC_HOSTNAME] ist der Computername für den Mastercomputer.
- [CLIENT_IP:PORT] ist die IP-Adresse und der Port des Clients, von dem die XML-Abfrage ausging.

- [ACCOUNT_NAME] ist der Kontoname, der für die Anmeldung bei der bereitgestellten FileMaker-Datenbank verwendet wird.
- [MODULE_TYPE] ist entweder: XML für Custom Web Publishing mit XML-Abfragen oder PHP für Custom Web Publishing mit PHP-Abfragen.
- [SEVERITY] ist entweder INFO, das eine informative Meldung angibt, oder ERROR, das eine Fehlermeldung angibt.
- [FM_ERRORCODE] ist die für eine Fehlermeldung zurückgegebene Fehlernummer. Die Fehlernummer kann ein Fehlercode für FileMaker-Datenbanken sein (siehe „Fehlercodenummern für FileMaker-Datenbanken“ auf Seite 101).
Zusätzlich kann die Fehlernummer eine HTTP-Fehlernummer sein, der die „HTTP:“-Zeichenfolge vorangestellt ist.
- [RETURN_BYTES] ist die Anzahl an Bytes, die die Anfrage zurückgibt.
- [MESSAGE] gibt zusätzliche Informationen zu dem Protokolleintrag.

Web Publishing Engine-Protokollmeldung

Beispiele

Die folgenden Beispiele zeigen die Arten der Meldungen, die in der wpe.log-Datei auftreten können:

Wenn die Web Publishing Engine startet oder anhält:

```
02.06.2019 15:15:31 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine started.
02.06.2019 15:46:52 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine stopped.
```

Erfolgreiche oder fehlgeschlagene XML-Abfrage-Anforderungen:

```
02.06.2019 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"
02.06.2019 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"
```

Scriptfehler:

```
02.06.2019 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: "Eigenes Dialogfeld anzeigen"
```

Änderungen an den Custom Web Publishing-Einstellungen:

```
09.06.2019 10:59:49 -0700 WPC_SERVER 192.168.100.101:0 jdoe - INFO
- - XML Web Publishing Engine is enabled.
```

Systemfehler:

```
02.06.2019 15:30:42 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML
ERROR - - Communication failed
```

Verwenden des Web-Server-Modul-Fehlerprotokolls

Wenn der Web-Server keine Verbindung zur Web Publishing Engine aufbauen kann, generiert das Web Server Modul eine Protokolldatei, die alle Fehler in Zusammenhang mit ihrer Ausführung aufzeichnet. Diese Datei heißt `web_server_module_log.txt` und befindet sich im Ordner „Logs“ im FileMaker Server-Ordner auf dem Web-Server-Host.

Verwenden der Tomcat-Protokolle

Wenn FileMaker Server ein von einem internen Web-Server-Fehler verursachtes Problem feststellt, finden Sie die Tomcat-Protokolle evtl. hilfreich. Die Tomcat-Protokolle befinden sich in der Web-Server-Komponente des FileMaker Server-Einsatzes:

- **Windows:**
 - `[Laufwerk]:\Programme\FileMaker\FileMaker Server\Admin\admin-master-tomcat\logs\`
wobei `[Laufwerk]` das Hauptlaufwerk ist, von dem das System startet.
 - `[Laufwerk]:\Programme\FileMaker\FileMaker Server\Web Publishing\publishing-engine\jwpc-tomcat\logs`
wobei `[Laufwerk]` das Hauptlaufwerk ist, von dem das System startet.
- **macOS:**
 - `/Library/FileMaker Server/Admin/admin-master-tomcat/logs/`
 - `/Library/FileMaker Server/Web Publishing/publishing-engine/jwpc-tomcat/logs`

Anhang A

Fehlercodes für Custom Web Publishing

Die Web Publishing Engine generiert Fehlercodes für Datenbank- und Abfrage-String-Fehler, die während einer XML-Daten-Abfrage auftreten können.

Eine Liste der aktuellen Fehlercodes finden Sie in der [FileMaker Knowledge Base](#).

Fehlercodenummern in XML-Format

Die Web Publishing Engine generiert für eine in XML-Format veröffentlichte Datenbank jedes Mal einen Fehlercode, wenn Daten angefordert werden. Diese Art von Fehlercodewert wird am Beginn des XML-Dokuments in das `<error code>`-Element für die `fmresultset`-Grammatik bzw. in das `<ERRORCODE>`-Element für die Grammatik `FMPXMLRESULT` oder `FMPXMLLAYOUT` eingefügt. Der Fehlercode 0 gibt an, dass kein Fehler aufgetreten ist.

Beispiele

Datenbankfehlercode in der `fmresultset`-Grammatik:

```
<?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>
```

Datenbankfehlercode in der `FMPXMLRESULT`-Grammatik:

```
<?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">
  ERRORCODE0/ERRORCODE
```

Als Entwickler der Custom Web Publishing-Lösung bleibt es Ihnen überlassen, den Wert des `<error code>`- bzw. `<ERRORCODE>`-Elements zu prüfen und geeignete Maßnahmen vorzusehen. Die Web Publishing Engine behandelt keine Datenbankfehler.

Fehlercodenummern für FileMaker-Datenbanken

Weitere Informationen zu FileMaker Pro Advanced-Fehlercodes finden Sie in der [FileMaker Pro Advanced Hilfe](#).

FileMaker Server stellt den Fehlercode 959 zur Verfügung, um anzugeben, dass eine Technologie deaktiviert wurde.

Wenn der Server-Administrator beispielsweise Custom Web Publishing über CLI deaktiviert, liefern XML-Abfragen den Fehlercode 959.

Beispiel:

Fehlercode 959 in der FMPXMLLAYOUT-Grammatik:

```
<FMPXMLLAYOUT>
  <ERRORCODE>959</ERRORCODE>
  <LAYOUT DATABASE="" NAME="" />
  <VALUELISTS />
</FMPXMLLAYOUT>
```

Index

Symbols

&-Zeichen, in PHP 74

A

Abfragen von Ausschnittfeldern 48

Abfragen von XML-Daten 26

Abfrage-Strings

Befehle und Parameter 39, 43

Datensätze in Ausschnitte einfügen 46

Datensätze in Ausschnitten bearbeiten 47

Richtlinien für 44

Variablenfelder, Syntax 48

Voll qualifizierter Feldname, Syntax 46

XML-Daten abfragen 39, 43

Abrufen

Layoutinformationen 52

Layoutnamen 51

Verfügbare Scriptnamen 52

Add, Befehl 76

add(), Methode 86

addSortRule(), Methode 84

Admin Console 15

Aktivieren von Custom Web Publishing in einer Datenbank 13

Anfangszeile, Einstellung 59

Anzahl an Zeilen, Einstellung 59, 89

ASCII-Zeichen, in XML-Dokumenten 39

Aus Werteliste, Überprüfung 90

Ausführen von Suchabfragen 84

Ausschnittdatensätze filtern, Einstellung 59, 89

Ausschnitte

Anfangszeile 59

Anzahl der Datensätze 59

Datensätze bearbeiten 47

Datensätze hinzufügen 46

Datensätze löschen 47

Datensätze sortieren 59

In PHP verwenden 81

Layout 59

Ausschnittfeld-Abfragen 59, 60

Authentifizierung von Webbenutzern 13

Authentifizierung Webbenutzer 13

Automatische Eingabe, Attribut 32

B

Bearbeiten von Datensätzen 76

Befehle, Abfrage *Siehe* Abfrage-Strings

Befehlszeilenschnittstelle (CLI) 15

Beispiele

Generierte FMPXMLLAYOUT-Grammatik 38

Generierte FMPXMLRESULT-Grammatik 35

Generierte fmresultset-Grammatik 33

Benutzernamen

Basisauthentifizierung für Webbenutzer 13

Für Custom Web Publishing definieren 13

Berechtigung, Zuweisen für Custom Web Publishing 13

Beschränken von Ausschnittfeld-Datensätzen 60

Bezugsmengenobjekt 81

C

clearSortRules(), Methode 84

CLI, Befehlszeilenschnittstelle 15

Client URL Library 66

commit(), Methode 75

Compound Find

Befehl 86

Beispiel 87

Compound-Suche

Abfrage-Befehl 50

Abfrage-Parameter 57

Containerfelder

Inhalt veröffentlichen 16

Mit extern gespeicherten Daten 17

Mit referenzierten Dateien 16

Progressive Downloads 18

URL-Syntax für Zugriff in XML-Lösungen 28

Wie Webbenutzer auf Daten zugreifen 18

createRecord(), Methode 75

cURL 66

Custom Web Publishing

Aktivieren in einer Datenbank 13

Definition 9

Erweitertes Zugriffsrecht für 13

Gastkonto 14

In Web Publishing Engine aktivieren 15

Mit PHP 11

Mit XML 11, 24

Neue Funktionen in 22

Plugins für Web Publishing-Lösungen 19

Scripts 20

Scripts verwenden 18

Statische IP-Adresse verwenden 23

Übersicht 9

Voraussetzungen für 22

Webbenutzer-Zugriff auf Lösungen 13

Zugriff von IP-Adressen in Web-Server beschränken 15

Custom Web Publishing Engine (CWPE) 25

D

Das von FileMaker Server zurückgegebene Unicode

-Datenformat 71

<datasource>-Element 31

Date, Feld 90

Datenbanken, veröffentlichte schützen 14

Datenbankfehlercodes 30, 100

Datenbankobjekt 75

Datensätze

- In PHP bearbeiten 76
 - In PHP duplizieren 76
 - In PHP erstellen 75
 - In PHP löschen 77
 - In PHP suchen 84
 - In XML bearbeiten 49
 - In XML duplizieren 49
 - In XML erstellen 51
 - In XML löschen 49
 - In XML suchen 50
 - In XML überspringen 63
- Datensatzobjekt 75
- Datumsdarstellung 88
- db (Abfrage-Parameter) 52
- dbnames (Abfrage-Befehl) 49
- delete (Abfrage-Befehl) 49
- Delete, Befehl 77
- delete.related query, Parameter 47
- delete(), Methode 77, 82
- Document Type Definitions (DTDs) 30, 34
- Dokumentation, FileMaker 8
- dup (Abfrage-Befehl) 49
- Duplicate, Befehl 76
- Duplizieren von Datensätzen 76
- Dynamische IP-Adresse 67

E

- edit (Abfrage-Befehl) 49
- Edit, Befehl 76
- Eindeutiger Wert, Überprüfung 90
- Elemente
 - Datenbankfehlercode 30
 - In FMPXMLLAYOUT-Grammatik 36
 - In FMPXMLRESULT-Grammatik 34
 - In fmresultset-Grammatik 31
- Ergebnismenge 88
- Erneut anmelden, Scriptschritt 14
- <Fehlercode>- und <ERRORCODE>-Elemente 100
- Erstellen eines Datensatzes
 - Mit PHP 75
 - Mit XML 51
- Erweitertes Zugriffsrecht für Custom Web Publishing 13
- Exportieren von XML-Daten 24
- Extensible Markup Language (XML) *Siehe XML*

F

- Fehler
 - Behandlung 93
 - Beschreibung 100
 - Datenbankfehlercode-Elemente 30
 - Datenbankfehlercodes 100
 - Protokolldateien für Web-Server 97
- Fehlerbehandlung 93
- Fehlerbehebung
 - Custom Web Publishing-Webseiten 95
 - XML-Dokumentzugriff 42

Felder

- Attribute 31
 - Ausschnitt 32
 - Ausschnitte 46
 - Bezugsfelder in PHP 81
 - Bezugsfelder in XML 32, 47
 - Container 16, 28, 34
 - Date 34, 37, 90
 - Formel 29, 30
 - Four-digit year 90
 - Maximum number of characters 89
 - Not empty 89
 - Number 34
 - Numeric only 89
 - Statistik 29, 30
 - Text 34
 - Time 34, 37, 90
 - Time of day 90
 - Timestamp 34, 37, 90
 - Variable 30
 - Voll qualifizierte Feldnamen 46
 - Wiederholfelder 28, 46
- Feldname (Abfrage-Parameter, nicht für Containerfeld) 53
- Feldnamen, voll qualifizierte Syntax 46
- field (Abfrage-Parameter, für Containerfeld) 53
- <field-definition>-Element 32
- fieldname.op (Abfrage-Parameter) 54
- FileMaker API für PHP 11
 - Manuelle Installation 67
 - Referenz 72
- FileMaker Pro Advanced, Unterschied zu Web Publishing Engine 24
- FileMaker Server Admin Console 15
- FileMaker Server-Dokumentation 8
- FileMaker WebDirect 9
- FileMaker, Klasse 73
- FileMaker, Klassenobjekte
 - Bezugsmenge 81
 - Datenbank 75
 - Datensatz 75
 - Definition 73
- FileMaker-Befehlsobjekte
 - Add 76
 - Compound Find 86
 - Delete 77
 - Duplicate 76
 - Edit 76
 - Find All 85
 - Find Any 85
 - Find, Befehl 84, 85
- Filtern von Ausschnittfeld-Datensätzen 59
- find (Abfrage-Befehl) 50
- Find All, Befehl 85
- Find Any, Befehl 85
- Find, Befehl 85
- findall (Abfrage-Befehl) 50
- findany (Abfrage-Befehl) 50
- findquery (Abfrage-Befehl) 50
- fmphp (Schlüsselwort zur Aktivierung von PHP-Publishing) 13

FMPXMLLAYOUT-Grammatik 24, 29, 36–38
 FMPXMLRESULT-Grammatik 24, 29, 34–35
 fmresultset-Grammatik 24, 29, 30–33
 fmsadmin-Gruppe 17
 fmxml (Schlüsselwort zur Aktivierung von XML-Publishing) 13, 26
 Four-digit year, Attribut 32
 Four-digit year, Feld 90

G

Gastkonto
 Aktivieren 14
 Deaktivieren 14
 Mit Custom Web Publishing 14
 getContainerData(), Methode 88
 getContainerDataURL(), Methode 88
 getDatabase(), Methode 80
 getErrors(), Methode 93
 getFetchCount(), Methode 88
 getField(), Methode 88
 getFieldAsTimestamp(), Methode 88
 getFields(), Methode 80, 88
 getFoundSetCount(), Methode 88
 getLayout(), Methode 80
 getMessage(), Methode 93
 getName(), Methode 80, 82
 getRange(), Methode 84
 getRecords(), Methode 88
 getRelatedSet(), Methode 81
 getRelatedSets(), Methode 81
 getValueListsTwoFields(), Methode 83
 getValueListTwoFields(), Methode 83
 Grammatiken für XML, Beschreibung 29

H

Herstellen einer Verbindung zu einer FileMaker-Datenbank mit PHP 75
 Homepage für PHP-Site 70
 hostspec-Eigenschaft für PHP 75
 HTML-Formulare für XML-Abfragen 26
 HTTPS 15
 mit Containerfeldern 18
 Protokoll in URL-Syntax 27
 Sicheres Verzeichnis für macOS 68
 HTTPS-Verzeichnis für PHP 70
 HTTP-Verzeichnis für PHP 70

I

Im Bereich, Überprüfung 90
 Importieren von XML-Daten 24
 Installation von FileMaker API für PHP 67
 isError(), Methode 93
 isValidatationError(), Methode 92

K

Kodieren
 PHP-Daten 71
 URLs 29
 XML-Daten 29, 39
 Konten und Berechtigungen
 Gastkonto 14
 Konten und Zugriffsrechte
 Für Custom Web Publishing aktivieren 13
 Scripts 19

L

Latin-1-Kodierung 71
 –lay (Abfrage-Parameter) 41, 55
 –lay.response (Abfrage-Parameter) 41, 55
 Layout bei Abfrage von XML-Daten angeben 41
 Layout wechseln für XML-Antwort 41
 –layoutnames (Abfrage-Befehl) 51
 Layouts
 Für XML-Antwort wechseln 41
 In PHP verwenden 80
 listFields(), Methode 80
 listLayouts(), Methode 80
 listRelatedSets(), Methode 80, 81
 listScripts(), Methode 77
 listValueLists(), Methode 80, 83
 –lop (Abfrage-Parameter) 56
 Löschen von Ausschnittdatensätzen 47
 Löschen von Datensätzen 77

M

macOS Server Admin 66
 Manuelle Installation von FileMaker API für PHP 67
 –max (Abfrage-Parameter) 56
 max-characters, Attribut 32
 Maximum number of characters, Feld 89
 max-repeat, Attribut 32
 <metadata>-Element 31
 Methoden
 add() 86
 addSortRule() 84
 clearSortRules() 84
 commit() 75
 createRecord() 75
 delete() 77, 82
 getContainerData() 88
 getContainerDataURL() 88
 getDatabase() 80
 getErrors() 93

getFetchCount() 88
 getField() 88
 getFieldAsTimestamp() 88
 getFields() 80, 88
 getFoundSetCount() 88
 getLayout() 80
 getMessage() 93
 getName() 80, 82
 getRange() 84
 getRecords() 88
 getRelatedSet() 81
 getRelatedSets() 81
 getValueListsTwoFields() 83
 getValueListTwoFields() 83
 isError() 93
 isValidationError() 92
 listFields() 80
 listLayouts() 80
 listRelatedSets() 80, 81
 listScripts() 77
 listValueLists() 80, 83
 newAddCommand() 76
 newCompoundFindCommand() 86
 newDeleteCommand() 77
 newDuplicateCommand() 76
 newEditCommand() 76
 newFindAllCommand() 85
 newFindAnyCommand() 85
 newFindCommand() 85
 newFindRequest() 86
 newPerformScriptCommand() 78
 newRelatedRecord() 82
 numErrors() 92, 93
 setLogicalOperator() 84
 setPreCommandScript() 78, 85
 setPreSortScript() 78, 85
 setProperty() 75
 setRange() 84
 setRelatedSetsFilters() 89
 setResultsLayout() 80
 setScript() 79, 85
 validate() 91
 MIME (Multipurpose Internet Mail Extensions-)-Typen 16
 -modid (Abfrage-Parameter) 57

N

name, Attribut 32
 Namensräume für XML 30
 Neue Funktionen in Custom Web Publishing 22
 -new (Abfrage-Befehl) 51
 newAddCommand(), Methode 76
 newCompoundFindCommand(), Methode 86
 newDeleteCommand(), Methode 77
 newDuplicateCommand(), Methode 76
 newEditCommand(), Methode 76
 newFindAllCommand(), Methode 85
 newFindAnyCommand(), Methode 85
 newFindCommand(), Methode 85
 newFindRequest(), Methode 86

newPerformScriptCommand(), Methode 78
 newRelatedRecord(), Methode 82
 Nicht genügend Arbeitsspeicher, Fehler 19
 Nicht leeres Feld 89
 not-empty, Attribut 32
 Numeric only, Feld 89
 numeric-only, Attribut 32
 numErrors(), Methode 92, 93
 Nummern für Datenbankfehlercodes 100

O

Online-Dokumentation 8
 Operatoren, Vergleich 54

P

Parameter, Abfrage *Siehe* Abfrage-Strings
 Passwort ändern (Script) 14
 Passwörter
 Basisauthentifizierung für Webbenutzer 13
 Für Custom Web Publishing definieren 13
 Kein Anmeldepasswort 14
 Passwort ändern (Script) 14
 PDFs 8
 PHP
 Custom Web Publishing, Beschreibung 11
 In einer Datenbank aktivieren 13
 Site-Homepage 70
 Unterstützte Version 67
 PHP-Version 66
 PHP-Vorteile 12
 Plugins 19
 Programmprotokoll 97
 Progressives Downloaden 18
 Protokolldateien 95
 Beschreibung 97
 Tomcat 99
 web_server_module_log.txt 99
 Web-Server-Zugriff 97

Q

-query (Abfrage-Parameter) 57

R

-recid (Abfrage-Parameter) 58
 Referenzinformationen 72
 Reihenfolge der XML-Abfrageverarbeitung 42
 <relatedset-definition>-Element 32
 -relatedsets.filter (Abfrage-Parameter) 59
 -relatedsets.max (Abfrage-Parameter) 60
 result, Attribut 32
 <resultset>-Element 32
 Rolle der Webbrowser in XML-Abfragen 25

- ## S
- SAT *Siehe* FileMaker Server Admin Console
 - Schlüsselwörter zur Aktivierung von Custom Web Publishing 13, 26
 - Schützen veröffentlichter Datenbanken 14
 - script (Abfrage-Parameter) 60
 - script.param (Abfrage-Parameter) 61
 - script.prefind (Abfrage-Parameter) 61
 - script.prefind.param (Abfrage-Parameter) 62
 - script.presort (Abfrage-Parameter) 62
 - script.presort.param (Abfrage-Parameter) 62
 - scriptnames (Abfrage-Befehl) 52
 - Scripts
 - In Custom Web Publishing 18
 - in PHP verwenden 77
 - In XML-Abfragen verwenden 26
 - Konten und Zugriffsrechte 19
 - Tipps und Überlegungen 19
 - Scriptschritte
 - Erneut anmelden 14
 - Passwort ändern 14
 - Script-Trigger 20
 - Server Admin-Tool. *Siehe* macOS Server Admin
 - Serveranforderungen 66
 - setLogicalOperator(), Methode 84
 - setPreCommandScript(), Methode 78, 85
 - setPreSortScript(), Methode 78, 85
 - setProperty(), Methode 75
 - setRange(), Methode 84
 - setRelatedSetsFilters(), Methode 89
 - setResultsLayout(), Methode 80
 - setScript(), Methode 79, 85
 - Sicherheit
 - Dokumentation 11
 - Konten und Passwörter 15
 - Richtlinien zum Schützen veröffentlichter Datenbanken 14
 - Zugriff von IP-Adressen beschränken 15
 - skip (Abfrage-Parameter) 63
 - sortfield (Abfrage-Parameter) 63
 - Sortieren von Ausschnittfeld-Datensätzen 59
 - sortorder (Abfrage-Parameter) 64
 - SSL (Secure Sockets Layer)-Verschlüsselung 15
 - Statische IP-Adresse 67
 - Statisches Veröffentlichen, Beschreibung 9
 - Stylesheets, testen 95
 - Suchbefehlsobjekte 84
- ## T
- Testen
 - Websites 95
 - XML-Ausgabe 96
 - Textkodierung
 - Generierte XML-Daten 29
 - URLs 29
 - Time of day, Feld 90
 - Time, Feld 90
 - time-of-day, Attribut 32
 - Timestamp, Feld 88, 90
 - Tomcat, Protokolldateien verwenden 99
 - type, Attribut 32
- ## U
- UAC. *Siehe* FileMaker Server Admin Console
 - Überprüfung
 - Befehle 89
 - Date 90
 - Datensätze 91
 - Felder 91
 - Four-digit year 90
 - Maximum number of characters 89
 - Not empty 89
 - Numeric only 89
 - Time 90
 - Time of day 90
 - Timestamp 90
 - Überprüfung durch Berechnung 90
 - Übersicht
 - Einführung in Custom Web Publishing 9
 - Schritte für XML-Datenzugriff 26
 - Überwachen von Webseiten 97
 - Unicode
 - In XML-Parsern verwendete Zeichen 39
 - Unix-Zeitstempel 88
 - URL-Syntax für
 - Containerobjekte in XML-Lösungen 28
 - XML-Abfragen 27
 - URL-Textkodierung 29
 - UTF-8-Format (Unicode Transformation 8 Bit) 29, 39
 - UTF-8-Kodierung 71
- ## V
- validate(), Methode 91
 - Variablenfelder
 - In Felddefinition 32
 - Syntax 48
 - Verarbeitung einer Ergebnismenge 88
 - Verarbeitung einer Web Publishing Engine-Anforderung 10
 - Verfügbare Datenbanklayouts 51
 - Verfügbare Scripts 52
 - Vergleich von XML-Grammatiken 29
 - Vergleichsoperatoren für Felder 54
 - Veröffentlichen im Web
 - Containerfelderobjekte 16
 - Datenbanken schützen 14
 - Datenbankfehlercodes 100
 - Mit XML 26
 - Verbindung zu Internet oder Intranet 23
 - Voraussetzungen für 22
 - Vertikaler Rollbalken, Einstellung 59, 89
 - view (Abfrage-Befehl) 52
 - Voll qualifizierter Feldname, Syntax 46

Vorabüberprüfung
 Befehle 89
 Date 90
 Datensätze 91
 Felder 91
 Four-digit year 90
 Maximum number of characters 89
 Not empty 89
 Numeric only 89
 Time 90
 Time of day 90
 Timestamp 90
Voraussetzungen für Custom Web Publishing 22
Vorhandener Wert, Überprüfung 90

W

Web Publishing Core, Illustration 25
Web Publishing Engine
 Anforderungsverarbeitung 10
 Beschreibung 10
 generierte Fehlercodes 100
 Programmprotokoll 97
 Vorteile 21
 XML-Daten generieren 25
 XML-Dokumente generieren 26
Web Server
 MIME-Typenunterstützung 16
 Protokolldateien 97
web_server_module_log.txt (Protokolldatei) 99
Webbenutzer
 Containerfelddaten verwenden 18
 Voraussetzungen für Zugriff auf Custom Web Publishing-
 Lösungen 23
 Zugriff auf geschützte Datenbanken 13
Web-Ordner, Containerfeldobjekte kopieren 16
Webseiten
 Testen 95
Web-Server
 Rolle in XML-Abfragen 25
Websites
 FileMaker Support-Seiten 8
 Mit FileMaker Web Publishing Engine erstellen 21
 Überwachen 97
Wertelisten
 in PHP verwenden 83
 in XML verwenden 36

X

XML
 Abfrage, Layout angeben 41
 Abfrage-Strings 39, 43
 Antwort, Layout wechseln 41
 Custom Web Publishing, Beschreibung 11
 Daten abfragen 26
 Document Type Definitions (DTDs) 30, 31, 34
 Fehlerbehebung beim Zugriff auf XML-Dokumente 42
 FMPXMLLAYOUT-Grammatik 36
 FMPXMLRESULT-Grammatik 34
 fmresultset-Grammatik 31
 <datasource>-Element 31
 <field-definition>-Element 32
 <metadata>-Element 31
 <relatedset-definition>-Element 32
 <resultset>-Element 32
 Grammatikvergleich 29
 Im UTF-8-Format kodiert 29, 39
 In einer Datenbank aktivieren 13
 Namensräume für 30
 Parser 26, 39
 Reihenfolge der Abfrageverarbeitung 42
 URL-Textkodierung 29
 Vorteile 12
 XML-1.0-Spezifikation 24
 XML-Daten aus Abfrage generieren 25
 Zusammenfassung der Schritte für Zugriff auf XML-
 Daten 26
 <xsl:stylesheet>-Element 96
 <xsl:template>-Element 96

Z

Zugriffsprotokolldateien für Web-Server, Beschreibung 97
Zugriffsrechte 14
Zusammenfassung der Schritte für XML-Datenzugriff 26