

FileMaker® Server 11

Custom Web Publishing
mit PHP



© 2007-2010 FileMaker, Inc. Alle Rechte vorbehalten.

FileMaker, Inc.

5201 Patrick Henry Drive

Santa Clara, California 95054, USA

FileMaker und Bento sind Marken von FileMaker, Inc., eingetragen in den USA und anderen Ländern. Das Dateiodner-Logo und das Bento-Logo sind Marken von FileMaker, Inc.

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. Die Danksagungen und Urheberrechtshinweise finden Sie im entsprechenden Dokument, das mit der Software geliefert wurde. 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.de.

Edition: 01

Inhalt

| | |
|--|----|
| Über dieses Handbuch | 7 |
| Kapitel 1 | |
| <i>Einführung in Custom Web Publishing</i> | |
| Über die Web Publishing Engine | 10 |
| Verarbeitung einer Web Publishing Engine -Anforderung | 10 |
| Custom Web Publishing mit PHP | 11 |
| Custom Web Publishing mit XML und XSLT | 11 |
| Vergleich von PHP mit XML und XSLT | 13 |
| Gründe für PHP | 13 |
| Gründe für XML und XSLT | 13 |
| Kapitel 2 | |
| <i>Erläuterung von Custom Web Publishing mit PHP</i> | |
| Wichtige Funktionen in Custom Web Publishing mit PHP | 15 |
| Anforderungen für Web-Publishing | 15 |
| Voraussetzungen für das Veröffentlichen einer Datenbank mit Custom Web Publishing | 16 |
| Voraussetzungen für Web-Benutzer zum Zugriff auf eine Custom Web Publishing-Lösung | 16 |
| Verbindung zum Internet oder zu einem Intranet | 16 |
| Manuelle Installation von FileMaker API for PHP | 17 |
| Die nächsten Schritte | 18 |
| Kapitel 3 | |
| <i>Vorbereiten von Datenbanken für Custom Web Publishing</i> | |
| Aktivierung von Custom Web Publishing mit PHP für Datenbanken | 19 |
| Erstellung von Layouts für Custom Web Publishing mit PHP | 19 |
| Schützen veröffentlichter Datenbanken | 20 |
| Zugriff auf eine geschützte Datenbank | 20 |
| Veröffentlichen des Inhalts von Medienfeldern im Web | 21 |
| Anzeige von Medienfelddaten durch Web-Benutzer | 22 |
| FileMaker-Scripts und Custom Web Publishing | 23 |
| Tipps und Überlegungen zu Scripts | 23 |
| Scriptverhalten in Custom Web Publishing-Lösungen | 24 |
| Script-Trigger in Custom Web Publishing-Lösungen | 25 |

Kapitel 4

Übersicht über Custom Web Publishing mit PHP

| | |
|---|----|
| Funktionsweise der Web Publishing Engine mit PHP-Lösungen | 27 |
| Allgemeine Schritte bei Custom Web Publishing mit PHP | 27 |
| Generieren von PHP-Websites mithilfe des FileMaker PHP-Site-Assistenten | 29 |
| Vor Verwendung des PHP-Site-Assistenten | 29 |
| Starten des PHP-Site-Assistenten | 29 |
| Verwendung von Sites, die mit dem PHP-Site-Assistenten generiert wurden | 31 |

Kapitel 5

Verwendung von FileMaker API for PHP

| | |
|--|----|
| Weitere Informationen | 33 |
| Referenz zu FileMaker API for PHP | 33 |
| FileMaker API for PHP-Lehrgang | 34 |
| Beispiele zu FileMaker API for PHP | 34 |
| Verwendung der Klasse FileMaker | 34 |
| FileMaker, Objekte der Klasse | 34 |
| FileMaker-Befehlsobjekte | 35 |
| Herstellen einer Verbindung zu einer FileMaker-Datenbank | 35 |
| Arbeiten mit Datensätzen | 36 |
| Erstellung eines Datensatzes | 36 |
| Duplizieren von Datensätzen | 36 |
| Bearbeiten von Datensätzen | 37 |
| Löschen von Datensätzen | 37 |
| Ausführen von FileMaker-Scripts | 37 |
| Abruf der Liste verfügbarer Scripts | 38 |
| Ausführen von FileMaker-Scripts | 38 |
| Ausführung eines Scripts vor der Ausführung eines Befehls | 38 |
| Ausführung eines Scripts vor der Sortierung einer Ergebnismenge | 38 |
| Ausführung eines Scripts nach der Erstellung einer Ergebnismenge | 38 |
| Ausführungsreihenfolge von Scripts | 39 |
| Arbeiten mit FileMaker-Layouts | 39 |
| Verwenden von Ausschnitten | 40 |
| Auflistung der in einem bestimmten Layout definierten Ausschnitte | 40 |
| Abrufen von Ausschnittnamen für ein bestimmtes Ergebnisobjekt | 40 |
| Abruf von Informationen über Ausschnitte für ein bestimmtes Layout | 40 |
| Abrufen von Informationen für einen bestimmten Ausschnitt | 40 |
| Abrufen des Tabellennamens für einen Ausschnitt | 40 |
| Abrufen der Ausschnittdatensätze für einen bestimmten Datensatz | 41 |
| Erstellen eines neuen Datensatzes in einem Ausschnitt | 41 |
| Löschen eines Datensatzes aus einem Ausschnitt | 41 |
| Verwenden von Wertelisten | 41 |
| Abrufen der Namen aller Wertelisten für ein bestimmtes Layout | 42 |
| Abrufen eines Arrays aller Wertelisten für ein bestimmtes Layout | 42 |

| | |
|--|----|
| Beziehen von Werten für eine benannte Werteliste | 42 |
| Ausführen von Suchabfragen | 43 |
| Der Befehl „Find All“ | 44 |
| Der Befehl „Find Any“ | 44 |
| Der Befehl „Find“ | 44 |
| Der Befehl „Compound Find“ | 45 |
| Verarbeitung der Datensätze in einer Ergebnismenge | 47 |
| Filtern von Ausschnittszeilen, die von Suchabfragen zurückgegeben wurden | 47 |
| Vorüberprüfung von Befehlen, Datensätzen und Feldern | 48 |
| Vorüberprüfung von Datensätzen in einem Befehl | 49 |
| Vorüberprüfung von Datensätzen | 50 |
| Vorüberprüfung von Feldern | 50 |
| Bearbeitung der Überprüfungsfehler | 50 |
| Fehlerbehandlung | 51 |

Kapitel 6

Einstellen, Testen und Überwachen einer Site

| | |
|---|----|
| Einstellen einer Custom Web Publishing-Site | 53 |
| Testen einer Custom Web Publishing-Site | 53 |
| Überwachen Ihrer Website | 54 |
| Verwenden der Web-Server-Zugriffs- und Fehlerprotokolle | 54 |
| Verwenden des Web Publishing Engine-Programmprotokolls | 55 |
| Verwenden des Web-Server-Modul-Fehlerprotokolls | 55 |
| Verwenden des internen Web Publishing Core-Zugriffsprotokolls | 55 |
| Fehlerbehebung für Ihre Site | 56 |

Anhang A

Fehlercodes für Custom Web Publishing mit PHP

| | |
|---|----|
| Fehlercodenummern für FileMaker-Datenbanken | 57 |
| Fehlercodenummern für PHP-Komponenten | 63 |

Index

65

Vorwort

Über dieses Handbuch

Dieses Handbuch setzt voraus, dass Sie mit PHP, der Entwicklung von Websites und dem Einsatz von FileMaker® Pro zur Erstellung von Datenbanken vertraut sind. Sie sollten mit den Grundlagen der Datenbankgestaltung von FileMaker Pro vertraut sein und die Konzepte von Feldern, Beziehungen, Layouts, Ausschnitten und Medienfeldern verstehen. Weitere Informationen zu FileMaker Pro finden Sie in der FileMaker Pro Hilfe.

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

- Voraussetzungen für die Entwicklung einer Custom Web Publishing-Lösung mit PHP
- Veröffentlichen Ihrer Datenbanken mit PHP
- Voraussetzungen für Web-Benutzer für den Zugriff auf eine Custom Web Publishing-Lösung
- Abruf von Daten aus von FileMaker Server bereitgestellten Datenbanken mithilfe des FileMaker API for PHP

Wichtig Sie können PDF-Dateien der FileMaker-Dokumentation von <http://www.filemaker.de/support/product/documentation.html> herunterladen. Aktualisierungen dieses Dokuments erhalten Sie ebenfalls auf der Website.

Die Dokumentation für FileMaker Server umfasst die folgenden Informationen:

| Weitere Informationen über | Siehe |
|--|---|
| Installieren und Konfigurieren von FileMaker Server | <i>FileMaker Server Einführung</i> <i>FileMaker Server Hilfe</i> |
| Instant Web Publishing | <i>FileMaker Instant Publishing Handbuch</i> |
| Custom Web Publishing mit PHP | <i>FileMaker Server Custom Web Publishing mit PHP</i> (dieses Handbuch) |
| Verwenden des PHP-Site-Assistenten | <i>PHP-Site-Assistent Hilfe</i> |
| Custom Web Publishing mit XML und XSLT | <i>FileMaker Server Custom Web Publishing mit XML und XSLT</i> |
| Verwenden des XSLT-Site-Assistenten | <i>XSLT-Site-Assistent Hilfe</i> |
| Installation und Konfiguration der ODBC- und JDBC-Treiber und Verwendung von ODBC und JDBC | <i>FileMaker ODBC- und JDBC-Handbuch</i> |
| Wie FileMaker Server Auto Update das aktuellste Plugin für FileMaker Pro-Datenbank-Clientcomputer herunterladen kann | <i>FileMaker-Handbuch zur Aktualisierung von Plugins</i> |

Kapitel 1

Einführung in Custom Web Publishing

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

Instant Web Publishing: Mit Instant Web Publishing können Sie Ihre Datenbank schnell und einfach im Web veröffentlichen. Sie müssen Ihre Datenbankdateien dazu nicht verändern oder zusätzliche Software installieren – jeder Benutzer mit einem kompatiblen Webbrowser und Zugriff auf das Internet bzw. ein Intranet kann eine Verbindung zu Ihrer Datenbank herstellen, um Datensätze anzuzeigen, zu bearbeiten, zu sortieren oder zu durchsuchen, wenn Sie ihm die entsprechenden Zugriffsrechte geben.

Für Instant Web Publishing muss der Host-Computer FileMaker Pro oder FileMaker Server ausführen. Die Benutzeroberfläche ähnelt der FileMaker Pro-Desktop-Anwendung. Die Web-Seiten und Formulare, mit denen der Web-Benutzer arbeitet, hängen von den in der FileMaker Pro-Datenbank definierten Layouts und Ansichten ab. Weitere Informationen finden Sie im *FileMaker Instant Web Publishing Handbuch*.

Statisches Veröffentlichen: Wenn sich Ihre Daten selten ändern oder 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-Datenbank, um eine Web-Seite zu erstellen, die Sie anhand von HTML weiter anpassen können. Die Web-Seite ändert sich nicht, wenn sich Informationen in Ihrer Datenbank ändern, und die Benutzer stellen keine Verbindung zu Ihrer Datenbank her. (Mit Instant Web Publishing werden die Daten in einem Webbrowser-Fenster jedes Mal aktiviert, wenn der Browser eine Anfrage an FileMaker Server sendet.) Weitere Informationen finden Sie im *FileMaker Instant Web Publishing Handbuch*.

Custom Web Publishing: Erscheinungsbild und Funktionalität Ihrer veröffentlichten Datenbank können Sie über die für FileMaker Server verfügbaren Technologien für Custom Web Publishing beeinflussen. Für FileMaker Server, der die veröffentlichten Datenbanken bereitstellt, muss weder FileMaker Pro installiert noch Custom Web Publishing verfügbar sein.

Mit Custom Web Publishing können Sie:

- Ihre Datenbank in andere Web-Sites 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 PHP: Verwenden Sie das FileMaker API for PHP, das eine objektorientierte PHP-Schnittstelle zu FileMaker Pro-Datenbanken bereitstellt, um Ihre FileMaker-Daten in einer PHP-Web-Anwendung zu integrieren. Sie können den PHP-Site-Assistenten verwenden, um eine komplette PHP-Website zu generieren, oder Ihre PHP-Web-Seiten selbst programmieren.
- Custom Web Publishing mit XML und XSLT:
 - Verwenden Sie XML Data Publishing, um FileMaker-Daten mit anderen Websites und Anwendungen auszutauschen.
 - Verwenden Sie Server-verarbeitete XSLT-Stylesheets, um beliebige Untergruppen von FileMaker-Daten in andere Websites, mit anderer Middleware und mit Spezialprogrammen zu integrieren. Sie können den XSLT-Site-Assistenten verwenden, um XSLT-Stylesheets zu generieren oder die Stylesheets selbst zu erstellen.

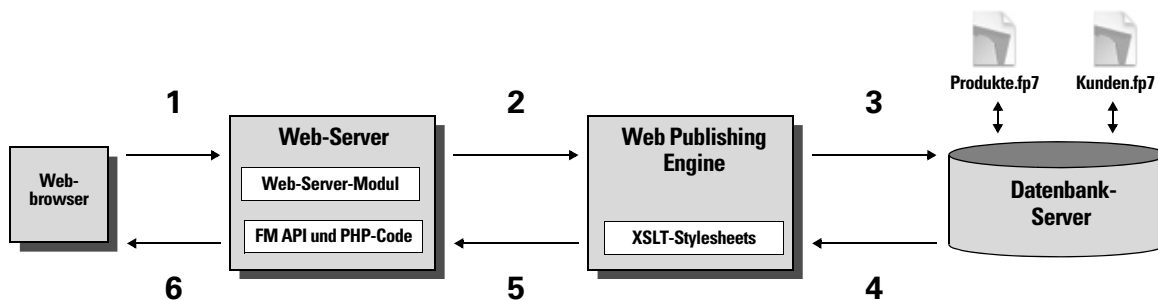
Über die Web Publishing Engine

Um Instant Web Publishing 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 Web-Benutzers, Ihrem Web-Server und FileMaker Server.

Custom Web Publishing mit XML und XSLT: Die Web Publishing Engine fungiert als XSLT-Prozessor und bietet Ausgaben als HTML, XML oder Text (z. B. vCards) auf dem Web-Server, der dann die Ausgabe für den Webbrowser liefert. Web-Benutzer 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-Query-Zeichenfolgenabfrage angibt. Der URL kann auf XML-Daten zugreifen oder auf ein XSLT-Stylesheet verweisen. Die Web Publishing Engine gibt die XML-Daten, die in der Query-Zeichenfolgenabfrage angegeben sind, oder das Ergebnis des referenzierten XSLT-Stylesheets zurück.

Custom Web Publishing mit PHP: Wenn ein Web-Benutzer 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 for 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.
 - Für PHP-Anforderungen antwortet die Web Publishing Engine auf die API-Anforderung.
 - Für XML-Anforderungen sendet die Web Publishing Engine die XML-Daten direkt an den Web-Server.
 - Für XSLT-Abfragen verwendet die Web Publishing Engine ein XSLT-Stylesheet, um die XML-Daten zu formatieren oder zu transformieren, und generiert die Ausgabe an den Web-Server als HTML-Seiten, XML-Dokument oder Text.
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.

Sehen Sie sich die Sicherheitsrichtlinien im *FileMaker Pro Benutzerhandbuch* an, das als PDF-Datei unter <http://www.filemaker.de/support/product/documentation.html> verfügbar ist.

Custom Web Publishing mit PHP

Das FileMaker API for PHP bietet eine objektorientierte PHP-Schnittstelle zu FileMaker-Datenbanken. Mithilfe des FileMaker API for PHP kann sowohl auf Daten als auch auf Logik, die in einer FileMaker Pro-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-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 Logikkonstrukte verwenden, um die Seitengenerierung, das Datenrouting oder den Workflow zu steuern. PHP bietet zudem Funktionen für Site-Administration und Sicherheit.

Zusätzlich können Sie den FileMaker PHP-Site-Assistenten verwenden, um PHP-Code zu erstellen, der alle erforderlichen Voraussetzungen und Funktionen beinhaltet, um korrekt auf die Daten in einer FileMaker Pro-Datenbank zuzugreifen. Der PHP-Site-Assistent generiert eine mehrseitige Website, die Web-Benutzer in die Lage versetzt, in einer Datenbank zu suchen, eine Liste von Datensätzen anzuzeigen, Datensätze zu durchblättern, zu sortieren, hinzuzufügen, zu bearbeiten, zu duplizieren, zu löschen und einen Statistikbericht anzuzeigen. FileMaker-Entwickler, die nur über geringe PHP-Erfahrung verfügen, können den PHP-Site-Assistenten verwenden, um eine ganze PHP-Website zu generieren. PHP-Entwickler, die nur über geringe Erfahrung mit FileMaker verfügen, können den PHP-Site-Assistenten verwenden, um die Objekte und Methoden des FileMaker API for PHP zu verstehen.

Custom Web Publishing mit XML und XSLT

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

FileMaker Custom Web Publishing mit XSLT ermöglicht das Umwandeln, Filtern oder Formatieren von XML-Daten für die Verwendung in Webbrowsern oder anderen Programmen. Sie können:

- ein XSLT-Stylesheet verwenden, um die Daten zwischen einer FileMaker XML-Grammatik und einer anderen XML-Grammatik für andere Programme oder andere Datenbanken umzuwandeln.

- die Daten filtern, indem Sie steuern, welche Datenbankfelder vom Stylesheet veröffentlicht werden.
- das Erscheinungsbild der Daten in einer Web-Seite formatieren und steuern, wie der Web-Benutzer mit den Daten interagiert.

Die Web Publishing Engine verwendet Ihre Stylesheets, um Daten aus einer FileMaker-Datenbank zu beziehen, wenn ein Web-Benutzer eine HTTP-Abfrage und einen URL sendet, der auf eines Ihrer XSLT-Stylesheets verweist. Die Web Publishing Engine verwendet ein Stylesheet, um die XML-Daten umzuwandeln und zu formatieren, und generiert die resultierende HTML-Seite, mit der der Web-Benutzer arbeiten kann.

Der FileMaker XSLT-Site-Assistent ist eine Anwendung, mit der Sie grundlegende XSLT-Stylesheets als Ausgangspunkt für Custom Web Publishing mit XSLT erstellen können. Der XSLT-Site-Assistent generiert Stylesheets für Seiten, die die Datenbank durchsuchen, einen Datensatz anzeigen, die Datensätze in der Datenbank auflisten, Datensätze sortieren, hinzuzufügen, bearbeiten, duplizieren, löschen und einen Statistikbericht anzeigen.

Vergleich von PHP mit XML und XSLT

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

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 for PHP kann sowohl auf Daten als auch auf Logik, die in einer FileMaker Pro-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 <http://php.net>.
- PHP ermöglicht den Zugriff auf eine große Vielzahl von Third-Party-Komponenten, die Sie in Ihre Lösungen integrieren können.

Gründe für XML und XSLT

- Die Syntax der FileMaker XML-Abfrageparameter ist für die Datenbankinteraktion konzipiert und vereinfacht die Lösungsentwicklung.
- XML und XSLT sind W3C-Standards.
- 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.
- Mit XSLT können Sie die XML-Ausgabe in strukturierte Textdokumente wie RSS, RTF oder vCard umwandeln.
- Sie können XSLT verwenden, um die XML-Ausgabe aus einer Grammatik in eine andere umzuwandeln.
- Vorlagen vereinfachen die Anwendung bedingter Formatierung auf variable Daten.
- Sie können FMPXMLRESULT-basierte Stylesheets für Custom Web Publishing und für den XML-Export aus FileMaker Pro-Datenbanken verwenden.
- FileMaker Server verwaltet die Verarbeitung von FileMaker-XSLT-Stylesheets und verhindert den unbefugten Zugriff auf Daten, die bei Verwendung von Client-seitigen XSLT-Stylesheets ungesichert sein könnten.

Hinweis Weitere Informationen über Custom Web Publishing mit XSLT finden Sie im Handbuch *FileMaker Server Custom Web Publishing mit XSLT*.

Kapitel 2

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 Web-Seiten-Layouts zu integrieren. Custom Web Publishing mit PHP stellt FileMaker API for 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 Web-Anwendungen, die die Open-Source-Scriptsprache PHP verwenden. Verwenden Sie die von FileMaker Server unterstützte Version PHP 5 oder Ihre eigene Version von PHP 5. (Wenn Sie Ihre eigene PHP-Version wählen, finden Sie genauere Informationen unter „Manuelle Installation von FileMaker API for PHP“ auf Seite 17.)
- Stellen Sie Datenbanken mit FileMaker Server bereit. FileMaker Pro ist für Custom Web Publishing nicht erforderlich, da FileMaker Server die Datenbanken bereitstellt.
- Verwenden Sie den PHP-Site-Assistenten zur Erstellung des PHP-Codes für Websites, die auf Daten in einer bereitgestellten FileMaker-Datenbank zugreifen. Informationen finden Sie unter „Generieren von PHP-Websites mithilfe des FileMaker PHP-Site-Assistenten“ auf Seite 29.
- 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 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. Weitere Informationen finden Sie unter „Schützen veröffentlichter Datenbanken“ auf Seite 20.
- Schreiben Sie PHP-Code, der komplexe Scripts mit mehreren Schritten ausführt. FileMaker unterstützt über 75 Scriptschritte in Custom Web Publishing. Weitere Informationen finden Sie unter „FileMaker-Scripts und Custom Web Publishing“ auf Seite 23.
- Schreiben Sie PHP-Code, der komplexe Suchabfragen ausführt.

Anforderungen für Web-Publishing

Dieser Abschnitt beschreibt, was für die Entwicklung einer Custom Web Publishing-Lösung mit PHP erforderlich ist, was Web-Benutzer 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 (Mac OS). Das FileMaker Web Server-Modul ist auf dem Web-Server installiert.
 - die FileMaker Web Publishing Engine
 - den FileMaker Datenbank-Server
- PHP ist auf dem Web-Server installiert. FileMaker Server kann die unterstützte Version PHP 5 installieren oder Sie können Ihre eigene Version verwenden. Die mindestens erforderliche Version von PHP unter Windows und Mac OS X v10.5 ist PHP 5.2.11. Die mindestens erforderliche Version von PHP unter Mac OS X v10.6 ist PHP 5.3.0. Weitere Informationen zu PHP erhalten Sie unter <http://php.net>. Die auf dem Web-Server installierte PHP-Version muss cURL-Funktionen (Client URL Library) unterstützen. Weitere Informationen zu cURL erhalten Sie unter <http://php.net/curl>.

Wichtig Wenn Sie die von FileMaker Server unterstützte Version PHP 5 installieren, wird es nicht im Mac OS X Server Admin-Tool angezeigt. Die Auflistungsfunktion wird nicht unterstützt. Wenn Sie das Mac OS X Server Admin-Tool für die Aktivierung von PHP verwenden, deaktivieren Sie die von FileMaker Server unterstützte Version PHP 5 und aktivieren Sie Ihre eigene PHP-Version.

- eine oder mehrere FileMaker Pro-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 Pro Einführung*.

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

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

- 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 Web-Benutzer 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 Web-Benutzern 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 Web-Benutzer 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 for PHP

Bei der Installation von FileMaker Server erhalten Sie die Möglichkeit, die von FileMaker unterstützte PHP-Version (PHP 5) zu installieren. Wenn bereits eine PHP-Engine installiert und konfiguriert ist und Sie nur FileMaker API for PHP hinzufügen möchten, installieren Sie die Klasse FileMaker API for PHP manuell, um Sie für Ihre PHP-Skripts zur Verfügung zu stellen.

Wenn Sie die von FileMaker unterstützte Version von PHP nicht installiert haben, 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 Daten und Uhrzeiten enthalten, installieren Sie das Pear-Date-Paket. Weitere Informationen finden Sie unter: <http://pear.php.net/package/date/>

Hinweis FileMaker Server wurde mit PHP Version 5.2.11 für Windows und für Mac OS X v10.5 und mit PHP Version 5.3.0 für Mac OS X v10.6 getestet. Verwenden Sie für optimale Ergebnisse die entsprechende PHP-Version.

Bereitstellung von FileMaker API for PHP für Ihre PHP-Skripts

Bei der Installation von FileMaker Server wurde das FileMaker API for 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 `<Laufwerk>` sich auf das Laufwerk bezieht, auf dem die Web-Server-Komponente Ihres FileMaker-Server-Einsatzes gespeichert ist.
- Für Apache (Mac OS): `/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 Root-Ordner des Web-Servers, in dem Ihre PHP-Skripts gespeichert sind.
 - Für IIS (Windows): `<Laufwerk>\inetpub\wwwroot`, wobei `<Laufwerk>` sich auf das Laufwerk bezieht, auf dem sich die Web Publishing Engine-Komponente Ihres FileMaker-Servers befindet.
 - Für Apache (Mac OS): `/Library/WebServer/Documents`
- eines der Verzeichnisse `include_path` in Ihrer PHP-Installation. Der Standardpfad für Mac OS X ist `/usr/lib/php`.

Die nächsten Schritte

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

- Verwenden Sie FileMaker Server Admin Console für die Aktivierung von Custom Web Publishing. Informationen hierzu finden Sie in der FileMaker Server Hilfe und im Handbuch *FileMaker Server Einführung*.
- Öffnen Sie in FileMaker Pro 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. Weitere Informationen finden Sie unter „Aktivierung von Custom Web Publishing mit PHP für Datenbanken“ auf Seite 19.
- Wie Sie mithilfe von FileMaker API for PHP auf Daten in FileMaker-Datenbanken zugreifen, erfahren Sie in Kapitel 5, „Verwendung von FileMaker API for PHP“.

Kapitel 3

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.

Aktivierung von Custom Web Publishing mit PHP für Datenbanken

Sie müssen Custom Web Publishing mit PHP in jeder Datenbank aktivieren, die Sie veröffentlichen wollen. Anderenfalls können Web-Benutzer Custom Web Publishing nicht für den Zugriff auf eine Datenbank verwenden, selbst wenn sie von FileMaker Server bereitgestellt wird, der zur Unterstützung einer Web Publishing Engine konfiguriert ist.

So aktivieren Sie Custom Web Publishing in einer Datenbank:

1. Öffnen Sie in FileMaker Pro die zu veröffentlichende Datenbank mit einem Konto, das über die Berechtigung für vollen Zugriff oder die Berechtigung „Erweiterte Zugriffsrechte verwalten“ verfügt.
2. Weisen Sie einer oder mehreren Berechtigungen das erweiterte Zugriffsrecht `fmphp` zu, um Custom Web Publishing mit PHP zu aktivieren.
3. Weisen Sie die Berechtigungen mit dem erweiterten Zugriffsrecht den entsprechenden Konten (zum Beispiel Admin und Gast) zu.

Wichtig 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 bestimmte nicht alphanumerische Zeichen wie ein Ausrufezeichen (!) oder ein Prozentzeichen (%). Doppelpunkte (:) sind nicht zulässig. Informationen über das Einrichten von Konten finden Sie in der FileMaker Pro Hilfe.

4. Stellen Sie mithilfe von FileMaker Server Admin Console sicher, dass die Bereitstellung der Datenbank richtig konfiguriert ist und FileMaker Server auf sie zugreifen kann. Informationen hierzu finden Sie in der FileMaker Server Hilfe.

Hinweis Da Custom Web Publishing mit PHP keine persistenten Datenbanksitzungen verwendet, können Verweise auf eine externe ODBC-Datenquelle im FileMaker Pro-Beziehungsdiagramm die Funktionen der PHP-Lösung einschränken. Wenn Ihre Datenbank auf Daten einer externen SQL-Datenquelle zugreift, können Sie die Datensätze der externen Tabelle unter Umständen nicht aktualisieren.

Erstellung von Layouts für Custom Web Publishing mit PHP

Custom Web Publishing mit PHP bietet keinen direkten Tabellenzugriff auf Daten in FileMaker Pro-Datenbanken, sondern verwendet die in den Datenbanken definierten Layouts. Zwar muss kein eigenes Layout für Custom Web Publishing mit PHP erstellt werden, es kann jedoch aus mehreren Gründen vorteilhaft sein, ein Layout speziell für eine PHP-Lösung zu entwickeln:

- Verbesserung der Leistung durch die Erstellung eines Layouts, das auf Felder, Beschriftungen und Ausschnitte beschränkt ist, die Sie für Ihre PHP-Lösung benötigen

- Vereinfachung des PHP-Codes durch einen geringeren Datenverarbeitungsaufwand, da weniger Felder vorhanden sind
- Trennung der Oberflächengestaltung von den Daten, sodass die Oberfläche an den Web-Benutzer angepasst werden kann

Schützen veröffentlichter Datenbanken

Mit Custom Web Publishing mit PHP können Sie den Zugriff auf Ihre veröffentlichten Datenbanken einschränken. Dabei stehen Ihnen folgende Methoden zur Verfügung:

- Fordern Sie für Datenbankkonten, die für Custom Web Publishing mit PHP verwendet werden, Passwörter an.
- Aktivieren Sie das erweiterte Zugriffsrecht für Custom Web Publishing mit PHP nur in den Berechtigungen, denen Sie den Zugriff ermöglichen möchten.
- Deaktivieren Sie Custom Web Publishing mit PHP für bestimmte Datenbanken, indem Sie das erweiterte Zugriffsrecht `fmphp` für sämtliche Berechtigungen in dieser Datenbank deaktivieren. Weitere Informationen finden Sie in der FileMaker Pro Hilfe.
- Aktivieren oder deaktivieren Sie Custom Web Publishing für alle Custom Web Publishing-Lösungen in der Web Publishing Engine mit der FileMaker Server Admin Console. Informationen hierzu finden Sie im Handbuch *FileMaker Server Einführung* und 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. Geben Sie beispielsweise an, dass nur Web-Benutzer 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.
- Verwenden Sie Secure Sockets Layer- (SSL) Verschlüsselung für die Kommunikation zwischen Ihrem Web-Server und den Webbrowsern. Die SSL-Verschlüsselung wandelt Informationen, die zwischen Servern und Clients übertragen werden, mithilfe von mathematischen Formeln in unverständliche Informationen um. Der englische Fachausdruck für diese Chiffrier-Algorithmen ist *Ciphers*. Diese Algorithmen nutzt der Empfänger, um mithilfe von Schlüsseln, den sogenannten Chiffrierschlüsseln, die Informationen wieder in verständliche Daten umzuwandeln. Informationen über das Aktivieren und Konfigurieren von SSL erhalten Sie in der Dokumentation zu Ihrem Web-Server.

Weitere Informationen zur Sicherung Ihrer Datenbank finden Sie im *FileMaker Pro Benutzerhandbuch*, das als PDF-Datei unter <http://www.filemaker.de/support/product/documentation.html> verfügbar ist.

Zugriff auf eine geschützte Datenbank

Wenn ein Web-Benutzer mit einer PHP-Lösung auf eine Datenbank zugreift, muss der PHP-Code die Zugangsdaten für die Datenbank über das FileMaker API for PHP bereitstellen. Wenn das Gastkonto für die Datenbank deaktiviert ist oder das erweiterte Zugriffsrecht `fmphp` nicht aktiviert ist, gibt das FileMaker API for PHP einen Fehler zurück und Ihr PHP-Code muss die Anmeldeinformationen für den Benutzer bereitstellen.

Der Lehrgang für das FileMaker API for PHP enthält ein Beispiel, das demonstriert, wie mithilfe der `setProperty()`-Methode ein Benutzername und ein Passwort für eine geschützte Datenbank festgelegt werden. Weitere Informationen finden Sie unter „FileMaker API for PHP-Lehrgang“ auf Seite 34.

Der PHP-Site-Assistent unterstützt zwei Optionen für den Zugriff auf geschützte Datenbanken:

- der PHP-Code kann den Web-Benutzer beim Zugriff auf die Site zur Authentifizierung auffordern.
- der PHP-Code kann den Datenbankkontonamen und das Passwort in den PHP-Dateien der Site speichern.

Genauere Informationen erhalten Sie in der PHP-Site-Assistent Hilfe.

Folgende Liste fasst die Abläufe zusammen, wenn Custom Web Publishing für den Zugriff auf eine Datenbank verwendet wird:

- Wenn einem Konto, das für Custom Web Publishing aktiviert wurde, kein Passwort zugewiesen wurde, muss die PHP-Lösung nur den Kontonamen bereitstellen.
- Wenn das Gastkonto deaktiviert ist, muss die PHP-Lösung einen Kontonamen und ein Passwort bereitstellen. Die PHP-Lösung kann entweder den Web-Benutzer nach dem Kontonamen und dem Passwort fragen oder den Kontonamen und das Passwort im PHP-Code speichern. Für das Konto muss das erweiterte Zugriffsrecht `fmphp` aktiviert sein.
- Wenn das Gastkonto aktiviert ist und das erweiterte Zugriffsrecht `fmphp` aktiviert ist:
 - Die PHP-Lösung muss Web-Benutzer beim Öffnen einer Datei nicht zur Eingabe des Kontonamens und Passworts auffordern. Alle Web-Benutzer werden automatisch mit dem Gastkonto angemeldet und erhalten die Gast-Zugriffsrechte.
 - Die Standardberechtigungen für Gastkonten umfassen den Nur-Lese-Zugriff. Sie können für dieses Konto die Standardberechtigungen einschließlich erweiterten Zugriffsrechten ändern. Weitere Informationen finden Sie in der FileMaker Pro Hilfe.
- Die PHP-Lösung kann den Scriptschritt „Erneut anmelden“ verwenden, um Benutzern zu ermöglichen, sich über ein anderes Konto anzumelden (um beispielsweise von einem Gastkonto in ein Konto mit mehr Berechtigungen zu wechseln). Weitere Informationen finden Sie in der FileMaker Pro Hilfe. Da PHP-Verbindungen jedoch keine persistenten Datenbanksitzungen verwenden, muss die PHP-Lösung den Kontonamen und das Passwort speichern, um sie für alle nachfolgenden Abfragen zu verwenden.

Hinweis Standardmäßig können Web-Benutzer ihr Kontopasswort nicht von einem Webbrowser aus ändern. Sie können diese Funktion über den Scriptschritt „Passwort ändern“ in Ihrer Datenbank aktivieren, damit Web-Benutzer ihr Passwort vom Browser aus ändern können. Weitere Informationen finden Sie in der FileMaker Pro Hilfe.

Veröffentlichen des Inhalts von Medienfeldern im Web

Der Inhalt eines Medienfelds, z. B. eine Bilddatei, kann entweder in einer FileMaker-Datenbank oder als Dateiverweis mit einem relativen Pfad gespeichert werden.

Verwendung des Inhalts eines Medienfelds in einer PHP-Lösung:

- Verwenden Sie die richtigen HTML-Tags, um das Web-kompatible Objekt anzugeben, das in dem Medienfeld enthalten ist, und erzeugen Sie eine URL-Zeichenfolge, die den Dateipfad für das Quellattribut des HTML-Tags wiedergibt.

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

- Verwenden Sie das FileMaker API for PHP, um das Datenbankobjekt mit den entsprechenden Zugangsdaten (Kontoname und Passwort) zu definieren, und verwenden Sie dann für den Abruf der Medienfelddaten die Methode `getContainerData()`.

```
$fm = & new FileMaker();
$fm->setProperty('database', $databaseName);
$fm->setProperty('username', $userName);
$fm->setProperty('password', $passWord);
echo $fm->getContainerData($_GET['-url']);
```

Der Lehrgang zum FileMaker API for PHP enthält weitere Beispiele zur Verwendung von Medienfeldern. Weitere Informationen finden Sie unter „FileMaker API for PHP-Lehrgang“ auf Seite 34.

Wenn ein Medienfeld 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 Medienobjektdateien im Web-Ordner des FileMaker Pro-Ordners.
2. Fügen Sie in FileMaker Pro die Objekte in das Medienfeld 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>:\inetpub\wwwroot`, wobei `<Laufwerk>` sich auf das Laufwerk bezieht, auf dem sich die Web Publishing Engine-Komponente Ihres FileMaker-Servers befindet.
 - Für Apache (Mac OS): `/Library/WebServer/Documents`

Hinweise

- Für Medienobjekte, die als Dateiverweise gespeichert sind, muss Ihr Web-Server so konfiguriert sein, dass er die MIME- (Multipurpose Internet Mail Extensions) 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.
- Alle QuickTime-Filme werden in Medienfeldern als Verweis gespeichert.

Anzeige von Medienfelddaten durch Web-Benutzer

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

- Web-Benutzer können keine Töne abspielen oder OLE-Objekte in einem Medienfeld anzeigen – stattdessen wird eine Grafik angezeigt.
- Web-Benutzer können den Inhalt eines Medienfelds nicht ändern oder ihm Inhalte hinzufügen. Web-Benutzer können Medienfelder nicht verwenden, um Daten in die Datenbank hochzuladen.
- Wenn Ihre Datenbank Grafiken in anderen Formaten als GIF oder JPEG enthält, erstellt die Web Publishing Engine ein temporäres JPEG-Bild, wenn die Grafikdaten von einem Webbrowser angefordert werden.
- Die Web Publishing Engine unterstützt kein Streaming von Filmen. Web-Benutzer müssen die gesamte Filmdatei herunterladen, bevor sie den Film ansehen können.

FileMaker-Scripts und Custom Web Publishing

Die Funktion „Scripts verwalten“ in FileMaker Pro kann sich häufig wiederholende Aufgaben automatisieren oder mehrere Aufgaben zusammenfassen. In Kombination mit Custom Web Publishing ermöglichen FileMaker-Scripts den Web-Benutzern, eine Reihe von Aufgaben durchzuführen. FileMaker-Scripts lassen zudem Aufgaben zu, die ansonsten nicht unterstützt werden, wie die Verwendung des Scriptschritts „Passwort ändern“, um Web-Benutzern die Passwortänderungen über einen Browser zu ermöglichen.

FileMaker unterstützt über 75 Scriptschritte in Custom Web Publishing. Um nicht unterstützte Scriptschritte anzuzeigen, wählen Sie im Dialogfeld „Script bearbeiten“ in FileMaker Pro die Option **Web Publishing** aus der Liste **Kompatibilität** anzeigen. Grau dargestellte Scripts werden nicht im Web unterstützt. Informationen über das Erstellen von Scripts finden Sie in der FileMaker Pro 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 24. 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.

Beachten Sie folgende Tipps und Überlegungen:

- 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 Web-Benutzer 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.
- Wählen Sie im Dialogfeld „Script bearbeiten“ Script mit vollen Zugriffsrechten ausführen, damit Scripts Aufgaben ausführen können, für die Sie dem Benutzer an sich keinen Zugang gewähren wollen. Beispielsweise können Sie Benutzern untersagen, Datensätze zu löschen, indem Sie ihre Konten und Berechtigungen beschränken, ihnen aber die Ausführung eines Scripts gestatten, das unter vordefinierten Bedingungen innerhalb eines Scripts bestimmte Arten von Datensätzen löscht.
- Wenn Ihre Scripts Schritte enthalten, die nicht unterstützt werden (z. B. Schritte, die nicht Web-kompatibel sind), verwenden Sie den Scriptschritt **AnwenderAbbruchZulassen setzen**, um festzulegen, wie nachfolgende Schritte gehandhabt werden.
 - Wenn der Scriptschritt **AnwenderAbbruchZulassen setzen** aktiviert (ein) ist, hindern nicht unterstützte Scriptschritte das Script an der Fortsetzung.
 - Wenn **AnwenderAbbruchZulassen setzen** ausgeschaltet ist, werden nicht unterstützte Scriptschritte übersprungen und das Script setzt die Ausführung fort.
 - Wenn dieser Scriptschritt nicht enthalten ist, werden Scripts ausgeführt, als ob die Funktion aktiv wäre, d. h., nicht unterstützte Scriptschritte stoppen Scripts.
- Einige Scripts, die mit einem Schritt von einem FileMaker Pro-Client aus funktionieren, benötigen eventuell einen zusätzlichen Schritt „Schreibe Änderung Datens./Abfrage“, um die Daten auf dem Host zu speichern. Da Web-Benutzer nicht über eine direkte Verbindung zum Host verfügen, werden sie nicht sofort benachrichtigt, wenn sich Daten ändern. Funktionen wie bedingte Wertelisten funktionieren z. B. nicht gleichermaßen für Web-Benutzer, da die Daten auf dem Host gespeichert werden müssen, bevor sich die Wirkung in einem Wertelistenfeld zeigt.

- Scripts, die Daten verändern, sollten den Schritt „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.
- 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 Hilfe.
- Nach der Konvertierung Ihrer Dateien sollten Sie jedes Script öffnen, das Web-Benutzer ausführen könnten. Wählen Sie dann Web Publishing aus der Liste Kompatibilität anzeigen im Fenster „Script bearbeiten“, um sicherzustellen, dass das Script mit Instant Web Publishing richtig ausgeführt wird.

Scriptverhalten in Custom Web Publishing-Lösungen

Die nachfolgenden Scriptschritte arbeiten im Web anders als in FileMaker Pro. Weitere Informationen zu allen Scriptschritten finden Sie in der FileMaker Pro Hilfe.

| Scriptschritt | Verhalten in Custom Web Publishing-Lösungen |
|-----------------------------------|--|
| Script ausführen | Scripts können nur dann in anderen Dateien ausgeführt werden, wenn die Dateien auf FileMaker Server bereitgestellt werden und Custom Web Publishing in den anderen Dateien aktiviert ist. |
| Programm beenden | Meldet Web-Benutzer ab, schließt alle Fenster, beendet aber den Webbrowser nicht. |
| AnwenderAbbruchZulassen setzen | Legt fest, wie nicht unterstützte Scriptschritte gehandhabt werden. Aktivieren, damit Scripts gestoppt werden; deaktivieren, damit nicht unterstützte Schritte übersprungen werden. Weitere Informationen finden Sie unter „Tipps und Überlegungen zu Scripts“ auf Seite 23. Hinweis Web-Benutzer können Custom Web Publishing-Scripts nicht abrechnen, aber diese Option ermöglicht, dass nicht unterstützte Scriptschritte die Fortsetzung des Scripts stoppen. |
| Fehlerrückmeldung setzen | Diese Option ist bei Custom Web Publishing immer aktiviert. Web-Benutzer können Custom Web Publishing-Scripts nicht abrechnen. |
| Scriptpause setzen | Diese Scriptschritte werden zwar in Custom Web Publishing unterstützt, sollten aber nicht verwendet werden. Wenn der Scriptschritt „Scriptpause setzen“ ausgeführt wird, hält das Script an. Nur ein Script, das erneut den Schritt „Scriptpause setzen“ enthält, kann die Ausführung fortsetzen. Wenn das Script im Pausezustand bleibt, bis die Sitzung endet, wird das Script nicht vollständig ausgeführt. |
| Datensätze sortieren | Sie müssen eine Sortierfolge mit dem Scriptschritt „Datensätze sortieren“ speichern, damit er in Custom Web Publishing ausgeführt wird. |
| URL öffnen | Dieser Scriptschritt hat keine Wirkung in einer Custom Web Publishing-Lösung. |
| Gehe zu Feld | Im Webbrowser können Sie „Gehe zu Feld“ nicht verwenden, um zu einem bestimmten Feld zu wechseln. Sie können diesen Scriptschritt jedoch in Kombination mit anderen Scriptschritten verwenden, um Aufgaben auszuführen. Beispielsweise können Sie zu einem Feld gehen und den Inhalt kopieren und dann zu einem anderen Feld gehen und den Wert einfügen. Um die Wirkung im Browser zu sehen, müssen Sie den Datensatz mit dem Scriptschritt „Datensatz bestätigen“ speichern. |
| Schreibe Änderung Datens./Abfrage | Sendet den Datensatz zur Datenbank. |

Script-Trigger in Custom Web Publishing-Lösungen

In FileMaker Pro 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. Wenn ein Custom Web Publishing-Benutzer zum Beispiel in ein Feld klickt, das einen Script-Trigger „BeiObjektBetreten“ besitzt, wird der Trigger nicht aktiviert. Wenn ein Script jedoch dazu führt, dass die Einfügemarke in das Feld wechselt, wird der Script-Trigger „BeiObjektBetreten“ aktiviert. Weitere Informationen zu Script-Trigger finden Sie in der FileMaker Pro Hilfe.

Kapitel 4

Übersicht über Custom Web Publishing mit PHP

Das FileMaker API for PHP unterstützt Sie bei der Integration von Daten aus FileMaker Pro-Datenbanken in PHP-Lösungen. Dieses Kapitel beschreibt die Funktionsweise von PHP mit der FileMaker Server Custom Web Publishing Engine und stellt ein Tool vor, das Ihnen den Einstieg in die Erstellung von PHP-Lösungen erleichtert: den FileMaker PHP-Site-Assistenten. Weitere Informationen über das FileMaker API for PHP finden Sie in Kapitel 5, „Verwendung von FileMaker API for 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. (Diese Komponenten können sich auf bis zu drei Rechner verteilen. Weitere Informationen hierzu finden Sie in der *FileMaker Server Einführung*.) FileMaker Server stellt die PHP-Lösung bereit, wenn Sie die PHP-Dateien auf dem Web-Server speichern, auf dem die PHP-Engine installiert ist.

- Wenn ein Web-Benutzer 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 for 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 Web-Benutzer an.

Allgemeine Schritte bei Custom Web Publishing mit PHP

Zusammenfassung der Schritte bei der Verwendung von Custom Web Publishing mit PHP:

1. Stellen Sie in Admin Console sicher, dass PHP Publishing aktiviert ist. Weitere Informationen finden Sie in der *FileMaker Server Einführung*.
2. Wählen Sie in der Admin Console den Bereich Datenbanken und stellen Sie sicher, dass für jede FileMaker-Datenbank, die Sie veröffentlichen, das erweiterte Zugriffsrecht `fmphp` für Custom Web Publishing mit PHP aktiviert ist.

Verwenden Sie, falls nötig, FileMaker Pro, um Custom Web Publishing für eine Datenbank zu aktivieren. Weitere Informationen finden Sie unter Kapitel 3, „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.

3. Verwenden Sie für die Erstellung Ihrer PHP-Lösung PHP-Authoring-Tools.

Sie können den FileMaker PHP-Site-Assistenten verwenden, um den grundlegenden PHP-Code für Ihre Site zu erstellen. Verwenden Sie diesen erstellten Code, ohne Änderungen vorzunehmen, oder verwenden Sie ihn als Grundlage für die Entwicklung einer komplexeren Site. Der vom PHP-Site-Assistenten entwickelte PHP-Code beinhaltet alle Voraussetzungen und Funktionen, um korrekt auf die Daten in einer FileMaker Pro-Datenbank zuzugreifen. Weitere Informationen finden Sie unter „Generieren von PHP-Websites mithilfe des FileMaker PHP-Site-Assistenten“ auf Seite 29.

4. Kopieren oder verschieben Sie Ihre Site-Verzeichnisstruktur und Ihre Dateien in den Root-Ordner des Web-Servers.

- Für IIS (Windows): <Laufwerk>:\inetpub\wwwroot, wobei <Laufwerk> sich auf das Laufwerk bezieht, auf dem sich die Web Publishing Engine-Komponente Ihres FileMaker-Servers befindet.
- Für Apache (Mac OS): /Library/WebServer/Documents

5. Wenn ein Datenbank-Medienfeld einen Dateiverweis anstelle der eigentlichen Datei speichert, muss das referenzierte Medienobjekt im FileMaker Pro-Web-Ordner 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.

Weitere Informationen finden Sie unter „Veröffentlichen des Inhalts von Medienfeldern im Web“ auf Seite 21.

6. Richten Sie unbedingt Sicherheitsmechanismen für Ihre Site oder Ihr Programm ein.

7. Testen Sie Ihre Site mit den Konten und Zugriffsrechten, die für Web-Benutzer definiert sind.

8. Geben Sie die Site frei und informieren Sie Benutzer. Der vom Web-Benutzer 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 Schritt 4 festgelegte Verzeichnisstruktur.

Wenn Ihr Web-Server beispielsweise 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 4 und PHP 5 verwenden die Kodierung Latin-1 (ISO-8859-1). FileMaker Server gibt Unicode-(UTF-8)-Daten zurück. Verwenden Sie FileMaker Server Admin Console, um die Standard-Zeichenkodierung für Ihre Site festzulegen. Für PHP-Sites können Sie entweder UTF-8 oder ISO-8859-1 angeben. Geben Sie die gleiche Einstellung für das charset-Attribut im <KOPF>-Bereich der PHP-Dateien Ihrer Site ein.

Generieren von PHP-Websites mithilfe des FileMaker PHP-Site-Assistenten

Der PHP-Site-Assistent unterstützt Sie bei der Erstellung einer einfachen PHP-basierten Website, um Daten in einer FileMaker Pro-Datenbank, die auf FileMaker Server bereitgestellt werden, abzurufen, zu bearbeiten und anzuzeigen. Der PHP-Site-Assistent führt Sie durch die Gestaltung und Konfiguration Ihrer Site und generiert danach den erforderlichen PHP-Code. Der PHP-Site-Assistent enthält vorkonfigurierte Präsentationsstile für die Gestaltung Ihrer Site.

Nach der Generierung des PHP-Codes für Ihre Site können Sie sie weiter mit jedem Standardtexteditor oder einem PHP-Authoring-Tool anpassen. Über den PHP-Site-Assistenten können Sie sich zudem selbst mit den grundlegenden Elementen des FileMaker API for PHP und den Grundlagen des Custom Web Publishing mit PHP vertraut machen.

Je nach den gewählten Optionen können Sie Seiten erstellen, auf denen Web-Benutzer folgende Aktionen durchführen können:

- Durchsuchen einzelner Datensätze
- Anzeige einer Liste mit allen Datensätzen in der Datenbank
- Durchsuchen der Datenbank und Anzeigen des Ergebnisses in einer Liste
- Sortieren von Datensätzen
- Hinzufügen von Datensätzen
- Bearbeiten und Duplizieren von Datensätzen
- Löschen von Datensätzen
- Anzeigen eines Statistikberichts

Zudem erstellen Sie eine Homepage, die mit den anderen erstellten Seiten der Website verknüpft ist. Genauere Informationen erhalten Sie in der PHP-Site-Assistent Hilfe.

Vor Verwendung des PHP-Site-Assistenten

Bevor Sie den PHP-Site-Assistenten zur Erstellung Ihrer Site verwenden:

1. Folgen Sie den Anleitungen in Kapitel 3, „Vorbereiten von Datenbanken für Custom Web Publishing“.
2. Stellen Sie mit FileMaker Server Admin Console sicher, dass der Web-Server und die Web Publishing Engine laufen. Informationen hierzu finden Sie in der FileMaker Server Hilfe.
3. Aktivieren Sie Custom Web Publishing mit PHP in der Web Publishing Engine, auf der der PHP-Code ausgeführt und getestet wird. Informationen hierzu finden Sie in der FileMaker Server Hilfe.
4. Wenn Sie über den PHP-Site-Assistenten eine Verbindung mit einer Datenbank herstellen, verwenden Sie ein Konto mit den Zugriffsrechten, die auch Web-Benutzern zugewiesen werden. Weitere Informationen zu Konten und Zugriffsrechten erhalten Sie in der FileMaker Pro Hilfe. Weitere Informationen zur Herstellung einer Verbindung mit einer Datenbank erhalten Sie in der PHP-Site-Assistent Hilfe.

Starten des PHP-Site-Assistenten

Der PHP-Site-Assistent ist auf dem Mastercomputer des FileMaker Server-Einsatzes installiert. Sie können den PHP-Site-Assistenten von jedem Computer aus verwenden, der Netzwerkzugriff auf den Mastercomputer besitzt. Wenn Sie den PHP-Site-Assistenten starten, verwendet FileMaker Server Java Web Start-Technologie, um den PHP-Site-Assistenten auf Ihren Client-Computer herunterzuladen. Der PHP-Site-Assistent läuft auf Ihrem Client-Computer und stellt nur dann eine Verbindung zu FileMaker Server her, wenn ein Zugriff auf die Datenbank erforderlich ist.

So starten Sie den PHP-Site-Assistenten:

1. Starten Sie den PHP-Site-Assistenten auf eine dieser Arten:

- Starten Sie einen Webbrowser und rufen Sie die Seite „FileMaker Server Web Publishing-Tools“ auf: <http://<host>:16000/tools> wobei <host> der Mastercomputer in Ihrem FileMaker Server-Einsatz ist. Klicken Sie auf der Seite "Tools" auf PHP-Site-Assistenten starten.
- Wählen Sie in FileMaker Pro Advanced Datei (Windows) bzw. Ablage (Mac OS) > Remote öffnen und öffnen Sie eine auf FileMaker Server bereitgestellte Datenbank, für die das erweiterte Zugriffsrecht fmpHP aktiviert ist. Wählen Sie Werkzeuge > PHP-Assistenten starten.
- Wählen Sie in FileMaker Server Admin Console Server > PHP-Site-Assistenten starten. Sie können auch auf das PHP-Site-Assistenten-Symbol in der Symbolleiste der FileMaker Server Admin Console klicken, um den PHP-Site-Assistenten zu starten.

FileMaker Server lädt die benötigten Dateien auf Ihren Computer herunter.

2. (Mac OS) Klicken Sie auf Vertrauen, um fortzufahren.

3. Klicken Sie im Dialogfeld „Verknüpfungen erstellen“ auf Ja, um eine Verknüpfung zum PHP-Site-Assistenten auf Ihrem Computer zu erstellen.

Windows: Die Verknüpfung wird auf dem Desktop gespeichert.

Mac OS: Geben Sie einen Namen und einen Speicherort für die Verknüpfung an und klicken Sie auf Speichern.

4. Sie können jetzt den PHP-Site-Assistenten verwenden. Vollständige Informationen zum Erstellen und Generieren von Sites erhalten Sie in der PHP-Site-Assistent Hilfe.



PHP-Site-Assistent, Fenster

Hinweise

- Um den PHP-Site-Assistenten von FileMaker Pro Advanced aus zu starten, müssen Sie die aktive Datenbankdatei auf FileMaker Server bereitstellen und das erweiterte Zugriffsrecht fmphp in einer Berechtigung für mindestens ein Konto aktivieren.
- Wenn sich die IP-Adresse des Mastercomputers in Ihrem FileMaker Server-Einsatz ändert, funktionieren die Verknüpfungen nicht mehr. Folgen Sie der Verfahrensweise oben, um neue Verknüpfungen zu erstellen.
- Sie können den PHP-Site-Assistenten starten, wenn Web Publishing ausgeschaltet ist. Wenn Sie aber versuchen, eine Verbindung zu einem Server herzustellen, bei dem Web Publishing ausgeschaltet ist, erhalten Sie einen Verbindungsfehler.

Verwendung von Sites, die mit dem PHP-Site-Assistenten generiert wurden

Sie können den PHP-Code, der vom PHP-Site-Assistenten generiert wurde, ohne Änderungen übernehmen oder Sie verwenden Ihre eigenen PHP-Authoring-Tools oder Texteditorprogramme, um die Site mit weiteren Funktionen und zusätzlichen Inhalten auszustatten. Sie können Ihre Site auch ohne den PHP-Site-Assistenten erstellen und die FileMaker API-Funktionen in Ihren PHP-Code integrieren, um auf Ihre FileMaker-Daten zuzugreifen. Weitere Informationen finden Sie unter Kapitel 5, „Verwendung von FileMaker API for PHP“.

Weitere Informationen zur Bereitstellung und Verwendung der mit dem PHP-Site-Assistenten erstellten Site finden Sie in Kapitel 6, „Einstellen, Testen und Überwachen einer Site“.

Kapitel 5

Verwendung von FileMaker API for PHP

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

Mit FileMaker API for PHP kann PHP-Code die Funktionen ausführen, die bereits in FileMaker Pro-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 for PHP wird in diesem Kapitel nicht erschöpfend behandelt, vielmehr werden die wichtigsten Objekte und Methoden vorgestellt.

Weitere Informationen

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

Falls Sie bereits eine PHP-Engine installiert und konfiguriert haben und nur FileMaker API for PHP hinzufügen möchten, finden Sie weitere Informationen unter „Manuelle Installation von FileMaker API for PHP“ auf Seite 17.

Referenz zu FileMaker API for PHP

Wenn FileMaker API for 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 (Mac OS): /Library/FileMaker Server/Documentation/PHP API Documentation/index.html

FileMaker API for PHP-Lehrgang

Wenn FileMaker API for PHP bereits installiert ist, finden Sie einen Lehrgang in der Web-Server-Komponente Ihres FileMaker-Server-Einsatzes.

- Für IIS (Windows): <Laufwerk>:\Programme\FileMaker\FileMaker Server\Examples\PHP\Tutorial wobei <Laufwerk> sich auf das Laufwerk bezieht, auf dem die Web-Server-Komponente Ihres FileMaker-Server-Einsatzes gespeichert ist.
- Für Apache (Mac OS): /Library/FileMaker Server/Examples/PHP/Tutorial

Kopieren Sie die PHP-Lehrgangsdateien in den Root-Ordner des Web-Servers, um sie bereitzustellen.

Beispiele zu FileMaker API for PHP

Wenn FileMaker API for PHP bereits installiert ist, finden Sie weitere Beispiele in der Web-Server-Komponente Ihres FileMaker-Server-Einsatzes.

- Für IIS (Windows): <Laufwerk>:\Programme\FileMaker\FileMaker Server\Examples\PHP\API Examples, wobei <Laufwerk> sich auf das Laufwerk bezieht, auf dem die Web-Server-Komponente Ihres FileMaker-Server-Einsatzes gespeichert ist.
- Für Apache (Mac OS): /Library/FileMaker Server/Examples/PHP/API Examples

Kopieren Sie die API-Beispieldateien in den Root-Ordner des Web-Servers, um sie bereitzustellen.

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-Datenbanken abrufen können.

| Klassenobjekt | Verwenden Sie das Objekt für |
|---------------------|---|
| FileMaker-Datenbank | die Definition von Datenbankeigenschaften die Verbindung mit einer FileMaker Pro-Datenbank den Abruf von Informationen über FileMaker API for 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 Ausschnittdatensätzen |
| 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:

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

Diese Befehle werden in den folgenden Abschnitten genauer beschrieben:

- „Arbeiten mit Datensätzen“ auf Seite 36
- „Ausführen von FileMaker-Scripts“ auf Seite 37
- „Ausführen von Suchabfragen“ auf Seite 43

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()`.

Beispiel: Herstellung einer Verbindung mit einem Server zum Abruf einer Datenbankliste

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

Beispiel: Herstellung einer Verbindung mit einer bestimmten Datenbank auf einem Server

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

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

Hinweis Die Eigenschaft „hostspec“ wird automatisch auf den Wert `http://localhost` gesetzt. Wenn die PHP-Engine auf demselben Rechner wie die Web-Server-Komponente des FileMaker Server-Einsatzes läuft, muss die Eigenschaft „hostspec“ nicht angegeben werden. Wenn sich die PHP-Engine auf einem anderen Rechner befindet, verwenden Sie die Eigenschaft „hostspec“, um den Speicherort der Web-Server-Komponente des FileMaker Server-Einsatzes anzugeben.

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-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();
```

- 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();
```

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 Datensatz-ID 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 Datensatz-ID 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 Datensatz-ID 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 Scriptschritten. Die Klasse `FileMaker` definiert mehrere Methoden, die es Ihnen ermöglichen, mit FileMaker-Scripts zu arbeiten, die in einer FileMaker Pro-Datenbank definiert sind. Weitere Informationen zu Web-kompatiblen Scriptschritten (Scriptschritte, die in einer Web-Lösung ausgeführt werden können) erhalten Sie unter „FileMaker-Scripts und Custom Web Publishing“ auf Seite 23.

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 35.)

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.

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. Weitere Informationen finden Sie unter „Der Befehl „Find““ auf Seite 44.

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. Weitere Informationen finden Sie unter „Der Befehl „Find““ auf Seite 44.

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.

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-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-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 Ausschnitt Datensätze 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-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()`. Der PHP-Site-Assistent unterstützt die Methode `getValueLists()` nicht.

Beziehen von Werten für eine benannte Werteliste

Verwenden Sie für ein bestimmtes Layoutobjekt die Methode `getValueListTwoFields()`, 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()`. Der PHP-Site-Assistent unterstützt die Methode `getValueList()` nicht.
- 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“ Weitere Informationen finden Sie unter „Der Befehl „Find All““ auf Seite 44.
- Befehl „Find Any“ Weitere Informationen finden Sie unter „Der Befehl „Find Any““ auf Seite 44.
- Befehl „Find“ Weitere Informationen finden Sie unter „Der Befehl „Find““ auf Seite 44.
- Befehl „Compound Find“ Weitere Informationen finden Sie unter „Der Befehl „Compound Find““ auf Seite 45.

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.

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

Beispiel – Suche nach einem Datensatz nach Feldnamen

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

Beispiel – 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.

Erstellung des Befehls „Command Find“:

- 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 `setOmit()`, um Datensätze in der Ergebnismenge einer bestimmten Suchabfrage anzugeben, die in der endgültigen Ergebnismenge nicht enthalten sein sollen.
- 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');
$findreq2->setOmit(true);

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

// 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 Instanziierung 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 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.

Filtern von Ausschnittzeilen, die von Suchabfragen zurückgegeben wurden

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 Ausschnitt Datensätze werden nicht vorsortiert.
 - Wenn Sie den Wert `layout` angeben, werden die im FileMaker Pro-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 maximale 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.

Vorabüberprüfung von Befehlen, Datensätzen und Feldern

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

Bevor Sie sich zu einer Vorabüberprüfung entscheiden, berücksichtigen Sie, wie viele Datenwerte der Web-Benutzer 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 T.M.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. 30.1.3030 ist beispielsweise ein gültiger Wert einer vierstelligen Jahreszahl. 31.4.2010 ist hingegen ein ungültiger Wert einer vierstelligen Jahreszahl, da der April nur 30 Tag hat. Die Datumsüberprüfung unterstützt Schrägstriche (/), umgekehrte Schrägstriche (\) und Bindestriche (-) als Trennzeichen. Die Zeichenfolge darf jedoch keine unterschiedlichen Trennzeichen enthalten. 30.1-2010 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). 30.1 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 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`-Objekt 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`-Objekt 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`-Objekt 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 `isValidationError()`, 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
$request =& $fm->newAddCommand('test', array('join' => 'added', 'maxchars' => 'abcx', 'field' => 'something',
'numericonly' => 'abc'));

//Alle Felder überprüfen
$result = $request->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";
    }
}
```

Ausgabe

```
Validation failed:
Feldname: numericonly
Error Code: 2
Wert: abc
Feldname: maxchars
Error Code: 3
Wert: abcx
```

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 mit PHP“.

Kapitel 6

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, kopieren oder verschieben Sie die erforderlichen Dateien an die richtigen Speicherorte auf dem bzw. den Servern.

So stellen Sie Ihre Site zum Testen ein und bereiten Sie dafür vor:

1. Folgen Sie den Anleitungen in Kapitel 3, „Vorbereiten von Datenbanken für Custom Web Publishing“.
2. Stellen Sie sicher, dass der Web-Server und die Web Publishing Engine laufen.
3. 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 auf dem Web-Server-Rechner:

- IIS (Windows): <Laufwerk>:\inetpub\wwwroot, wobei <Laufwerk> sich auf das Laufwerk bezieht, auf dem sich die Web Publishing Engine-Komponente Ihres FileMaker-Servers befindet.
 - Apache (Mac OS): /Library/WebServer/Documents
4. Wenn nicht bereits geschehen, kopieren oder verschieben Sie referenzierte Medienobjekte in das richtige Verzeichnis auf dem Web-Server-Rechner.

Wenn ein Datenbank-Medienfeld einen Dateiverweis anstelle der eigentlichen Datei speichert, wird das referenzierte Medienobjekt im FileMaker Pro-Web-Ordner gespeichert, wenn der Datensatz erstellt oder bearbeitet wird. Um Ihre Site einzustellen, kopieren oder verschieben Sie die referenzierten Medien in einen Ordner mit dem gleichen relativen Speicherort im Root-Ordner der Web-Server-Software.

Weitere Informationen finden Sie unter „Veröffentlichen des Inhalts von Medienfeldern im Web“ auf Seite 21.

5. Beginnen Sie mit dem Test Ihrer Website.

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 nicht autorisierte 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 23 für Informationen zur Gestaltung Web-freundlicher Scripts).
- Testen Sie Ihre Site mit unterschiedlichen Betriebssystemen und Webbrowsern.

Hinweis Wenn Web-Server, Web Publishing Engine und Datenbank-Server auf einem einzigen Computer installiert sind, können Sie Ihre Site ohne Netzwerkverbindung anzeigen und testen. Verschieben Sie Ihre Website-Dateien in das entsprechende Verzeichnis auf dem Rechner und geben Sie folgende URLs in Ihren Browser ein:

- `http://localhost/<site_path>`
- `http://127.0.0.1/<site_path>`

Dabei ist `<site_path>` der relative Pfad zur Homepage Ihrer Site.

Überwachen Ihrer Website

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

- Web-Server-Zugriff und Fehlerprotokolle
- Web Publishing Engine-Programmprotokoll
- Web-Server-Modul-Fehlerprotokoll
- interne Web Publishing Core-Zugriffsprotokolle

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 (nur Mac OS): 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 über diese Protokolldateien finden Sie in der Dokumentation zu Ihrem 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-Programmprotokolls

Standardmäßig generiert die Web Publishing Engine eine Programmprotokolldatei, die eine Aufzeichnung von Web Publishing Engine-Fehler-, Script- und Benutzerprotokollinformationen enthält. Die Protokolldatei heißt `pe_application_log.txt` und befindet sich in der Web Publishing Engine-Komponente Ihres FileMaker Server-Einsatzes:

- IIS (Windows):
`<Laufwerk>\Programme\FileMaker\FileMaker Server\Logs\pe_application_log.txt`
 Dabei ist `<Laufwerk>` das Hauptlaufwerk, von dem Ihr System startet.
- Apache (Mac OS): `/Library/FileMaker Server/Logs/pe_application_log.txt`

Die Datei `pe_application_log.txt` wird generiert, wenn beliebige der folgenden Protokolloptionen in der Web Publishing Engine aktiviert sind:

| Protokollierungsoption aktiviert | In <code>pe_application_log.txt</code> aufgezeichnete Informationen |
|----------------------------------|---|
| Fehlerprotokollierung | Ungewöhnliche Web Publishing Engine-Fehler, die aufgetreten sind. Dem Web-Benutzer häufig gemeldete Fehler wie „Datenbank nicht geöffnet“ werden nicht aufgezeichnet. |
| Scriptprotokollierung | Fehler, die erzeugt werden, wenn Web-Benutzer Scripts ausführen. Sie listet z. B. Schrittschritte auf, die übersprungen wurden, weil sie nicht Web-kompatibel sind. |

Beide Protokollierungsoptionen sind standardmäßig aktiviert. Informationen zur Einstellung dieser Optionen über Admin Console finden Sie in der *FileMaker Server Einführung*.

Hinweis Die Einträge im Programmprotokoll werden nicht automatisch gelöscht, daher kann die Datei mit der Zeit sehr umfangreich werden. Um Festplattenspeicherplatz auf Ihrem Host-Computer zu sparen, empfiehlt sich ein regelmäßiges Archivieren der Programmprotokolldatei.

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 Protokolldatei heißt `web_server_module_log.txt` und befindet in der Web-Server-Komponente Ihres FileMaker Server-Einsatzes:

- IIS (Windows): `<Laufwerk>\Programme\FileMaker\FileMaker Server\Logs\web_server_module_log.txt`
 Dabei ist `<Laufwerk>` das Hauptlaufwerk, von dem Ihr System startet.
- Apache (Mac OS): `/Library/FileMaker Server/Logs/web_server_module_log.txt`

Verwenden des internen Web Publishing Core-Zugriffsprotokolls

Standardmäßig generiert die Web Publishing Core-Komponente der Web Publishing Engine eine Protokolldatei für den Zugriff auf den Web Publishing Core. Die Protokolldatei heißt `wpc_access_log.txt`. Sie enthält eine Aufzeichnung aller Endanwenderabfragen zur Generierung von Web Publishing-Ausgaben und zur Verwendung von FileMaker Server Instant Web Publishing. Diese Abfragen werden vom Web-Server direkt an den Web Publishing Core geleitet.

Da das FileMaker API for PHP HTTP POST für den Zugriff auf den Web Publishing Core verwendet, zeichnet die Protokolldatei keine Details zu den PHP-Abfragen auf. Die Protokolldatei wird herangezogen, um zu erfahren, wann Benutzer Abfragen eingegeben haben.

Diese Datei befindet sich im folgenden Verzeichnis in der Web Publishing Engine-Komponente Ihres FileMaker Server-Einsatzes:

- IIS (Windows): <Laufwerk>\Programme\FileMaker\FileMaker Server\Logs\wpc_access_log.txt
Dabei ist <Laufwerk> das Hauptlaufwerk, von dem Ihr System startet.
- Apache (Mac OS): /Library/FileMaker Server/Logs/wpc_access_log.txt

Fehlerbehebung für Ihre Site

Wenn Sie Probleme bei der Anzeige oder Verwendung Ihrer Site haben, überprüfen Sie Folgendes:

- Die erweiterten Zugriffsrechte in der Datenbank müssen für Custom Web Publishing mit PHP eingestellt und einem Benutzerkonto zugewiesen sein. Weitere Informationen finden Sie unter „Aktivierung von Custom Web Publishing mit PHP für Datenbanken“ auf Seite 19.
- Die Datenbank muss über FileMaker Server bereitgestellt und geöffnet sein. Weitere Informationen finden Sie in der FileMaker Server Hilfe.
- Prüfen Sie, ob der verwendete Datenbankkontoname und ggf. Ihr Passwort korrekt sind.
- Stellen Sie sicher, dass Web-Server und Web Publishing Engine laufen.
- PHP-Publishing muss in der Web Publishing Engine aktiviert sein.
 - Öffnen Sie einen Browser mit der Seite FileMaker Server Technologietests:
http://<Server>:16000/test
wobei <Server> der Rechner ist, auf dem sich FileMaker Server befindet.
 - Klicken Sie auf den Link PHP Custom Web Publishing testen, um eine PHP-Seite zu öffnen, die auf die Testdatenbank FMServer_Sample zugreift.

Weitere Informationen finden Sie im Handbuch *FileMaker Server Einführung* und in der FileMaker Server Hilfe.

Anhang A

Fehlercodes für Custom Web Publishing mit PHP

Die Web Publishing Engine unterstützt zwei Arten von Fehlercodes, die für Custom Web Publishing auftreten können:

- Datenbank- und Datenabfragefehler. Die Web Publishing Engine generiert einen Fehlercode, wenn Daten von einer veröffentlichten Datenbank angefordert werden. Das FileMaker API for PHP gibt diesen Fehlercode als FileMaker_Error-Objekt zurück. Weitere Informationen finden Sie im nächsten Abschnitt, „Fehlercodenummern für FileMaker-Datenbanken“.
- PHP-Fehler. Diese Fehler werden von PHP-Komponenten, einschließlich des cURL-Moduls, generiert und zurückgegeben. Weitere Informationen finden Sie unter „Fehlercodenummern für PHP-Komponenten“ auf Seite 63.

Fehlercodenummern für FileMaker-Datenbanken

Als Entwickler der Custom Web Publishing-Lösung obliegt es Ihnen, den Wert des Fehlercodes zu prüfen und geeignete Maßnahmen zu ergreifen. Die Web Publishing Engine behandelt keine Datenbankfehler.

| Fehlernummer | Beschreibung |
|--------------|---|
| -1 | Unbekannter Fehler |
| 0 | Kein Fehler |
| 1 | Aktion durch Benutzer abgebrochen |
| 2 | Speicherfehler |
| 3 | Befehl nicht verfügbar (z. B. falsches Betriebssystem, falscher Modus etc.) |
| 4 | Befehl unbekannt |
| 5 | Befehl ungültig (z. B. ist für den Scriptschritt „Feldwert setzen“ keine Formel angegeben). |
| 6 | Datei ist schreibgeschützt |
| 7 | Speicherüberlauf |
| 8 | Leeres Ergebnis |
| 9 | Ungenügende Berechtigungen |
| 10 | Angeforderte Daten fehlen |
| 11 | Name ist nicht gültig |
| 12 | Name existiert bereits |
| 13 | Datei oder Objekt ist in Gebrauch |
| 14 | Außerhalb des gültigen Bereichs |
| 15 | Teilen durch null nicht möglich |
| 16 | Operation fehlgeschlagen, Anfrage wiederholen (z. B. eine Benutzeranfrage) |
| 17 | Konvertierung von fremdem Zeichensatz in UTF-16 fehlgeschlagen |
| 18 | Client muss Kontoinformationen liefern, um fortzufahren |
| 19 | Zeichenfolge enthält andere Zeichen als A-Z, a-z, 0-9 (ASCII) |

| Fehlernummer | Beschreibung |
|---------------------|---|
| 20 | Befehl oder Vorgang durch ausgelöstes Script abgebrochen |
| 100 | Datei fehlt |
| 101 | Datensatz fehlt |
| 102 | Feld fehlt |
| 103 | Beziehung fehlt |
| 104 | Script fehlt |
| 105 | Layout fehlt |
| 106 | Tabelle fehlt |
| 107 | Index fehlt |
| 108 | Werteliste nicht vorhanden |
| 109 | Berechtigung fehlt |
| 110 | Bezugstabellen fehlen |
| 111 | Feldwiederholung ist ungültig |
| 112 | Fenster fehlt |
| 113 | Funktion fehlt |
| 114 | Dateiverweis fehlt |
| 115 | Menüset fehlt |
| 116 | Layoutobjekt fehlt |
| 117 | Datenquelle fehlt |
| 130 | Dateien sind beschädigt oder fehlen und müssen neu installiert werden |
| 131 | Dateien des Sprachpakets fehlen (z. B. Vorlagendateien) |
| 200 | Zugriff auf Datensatz verweigert |
| 201 | Feld kann nicht geändert werden |
| 202 | Zugriff auf Feld verweigert |
| 203 | Keine zu druckenden Datensätze in der Datei bzw. Passwort erlaubt kein Drucken |
| 204 | Kein Zugriff auf Feld(er) in Sortierfolge |
| 205 | Benutzer hat keine Zugriffsrechte, um neue Datensätze zu erstellen; Import überschreibt bestehende Daten |
| 206 | Benutzer hat keine Zugriffsrechte, um das Passwort zu ändern, oder die Datei ist schreibgeschützt |
| 207 | Benutzer hat nicht genügend Zugriffsrechte, um das Datenbankschema zu ändern, oder die Datei ist schreibgeschützt |
| 208 | Passwort enthält zu wenige Zeichen |
| 209 | Neues Passwort muss sich vom bestehenden unterscheiden |
| 210 | Benutzerkonto ist inaktiv |
| 211 | Passwort ist abgelaufen |
| 212 | Ungültiges Benutzerkonto und/oder Passwort. Versuchen Sie es erneut |
| 213 | Benutzerkonto und/oder Passwort existieren nicht |
| 214 | Zu viele Anmeldeversuche |

| Fehlernummer | Beschreibung |
|---------------------|--|
| 215 | Administratorrechte können nicht dupliziert werden |
| 216 | Gastkonto kann nicht dupliziert werden |
| 217 | Benutzer hat nicht genügend Zugriffsrechte, um Administratorkonto zu ändern |
| 300 | Datei ist geschützt oder in Gebrauch |
| 301 | Datei ist blockiert durch anderen Anwender |
| 302 | Tabelle ist blockiert durch anderen Anwender |
| 303 | Datenbankschema ist blockiert durch anderen Anwender |
| 304 | Layout ist blockiert durch anderen Anwender |
| 306 | Datensatzänderungs-ID stimmt nicht überein |
| 400 | Suchkriterien sind leer |
| 401 | Kein Datensatz entspricht der Abfrage |
| 402 | Kein Abgleichsfeld für eine Referenz |
| 403 | Maximales Datensatzlimit für FileMaker Pro-Demo wird überschritten |
| 404 | Ungültige Sortierfolge |
| 405 | Angegebene Datensatzzahl übersteigt die Anzahl der ausschließbaren Datensätze |
| 406 | Ungültige Kriterien für Ersetzen/Neunummerierung |
| 407 | Ein oder beide Gruppierfeld(er) fehlen (ungültige Beziehung) |
| 408 | Angegebenes Feld hat ein Datenformat, das diesem Befehl nicht entspricht |
| 409 | Ungültige Importfolge |
| 410 | Ungültige Exportfolge |
| 412 | Falsche Version von FileMaker Pro verwendet, um die Datei wiederherzustellen |
| 413 | Angegebenes Feld hat ungeeigneten Feldtyp |
| 414 | Layout kann das Ergebnis nicht anzeigen |
| 415 | Ein oder mehrere erforderliche Bezugsdatensätze sind nicht verfügbar |
| 416 | Primärschlüssel erforderlich für Datenquelltablelle |
| 417 | Die Datenbank ist keine unterstützte Datenquelle |
| 500 | Datumswert entspricht nicht den Überprüfungskriterien |
| 501 | Zeitwert entspricht nicht den Überprüfungskriterien |
| 502 | Zahlenwert entspricht nicht den Überprüfungskriterien |
| 503 | Feldwert entspricht nicht der Bereichsüberprüfung |
| 504 | Feldwert entspricht nicht der Eindeutigkeitsüberprüfung |
| 505 | Feldwert existiert nicht in der Datenbank und entspricht nicht der Existenzüberprüfung |
| 506 | Feldwert entspricht nicht der Überprüfung nach Bestandteil einer Werteliste |
| 507 | Feldwert entspricht nicht der Überprüfung durch Berechnung |
| 508 | Ungültiger Wert wurde in Suchenmodus eingegeben |
| 509 | Feld verlangt gültigen Wert |
| 510 | Bezugswert ist leer oder nicht verfügbar |

| Fehlernummer | Beschreibung |
|---------------------|---|
| 511 | Der Wert im Feld überschreitet die maximale Anzahl der zulässigen Zeichen |
| 512 | Datensatz wurde bereits durch anderen Anwender geändert |
| 600 | Druckerfehler aufgetreten |
| 601 | Kombination von Kopf- und Fußbereich übersteigt eine Seitenlänge |
| 602 | Datenbereich passt für aktuelle Spalteneinstellung nicht auf eine Seite |
| 603 | Verbindung zum Drucker getrennt |
| 700 | Datei hat falschen Dateityp für Import |
| 706 | EPSF-Datei hat keine Übersichtsgrafik |
| 707 | Grafikfilter nicht vorhanden |
| 708 | Dateiimport nicht möglich bzw. Farbmonitor für Import erforderlich |
| 709 | Import des QuickTime-Films fehlgeschlagen |
| 710 | QuickTime-Dateiverweis konnte nicht aktualisiert werden, da für Datenbankdatei nur Lesezugriff besteht |
| 711 | Importfilter nicht vorhanden |
| 714 | Zugriffsrechte reichen für diesen Befehl nicht aus |
| 715 | Benannter Bereich oder Tabellenblatt von Excel konnte nicht gefunden werden |
| 716 | Eine SQL Anfrage mit DELETE, INSERT oder UPDATE ist nicht für ODBC Import erlaubt |
| 717 | Zum Fortsetzen des Imports bzw. Exports sind nicht genügend XML/XSL-Daten vorhanden |
| 718 | XML-Parsingfehler (von Xerces) |
| 719 | Fehler beim Transformieren von XML mit XSL (von Xalan) |
| 720 | Fehler beim Export: Das vorgesehene Format unterstützt keine Wiederholfelder |
| 721 | Im Parser oder Transformer ist ein unbekannter Fehler aufgetreten |
| 722 | Daten können nicht in eine Datei importiert werden, die keine Felder hat |
| 723 | Sie sind nicht berechtigt, Datensätze in der Zieltabelle hinzuzufügen oder zu ändern |
| 724 | Sie sind nicht berechtigt, Datensätze in der Zieltabelle hinzuzufügen |
| 725 | Sie sind nicht berechtigt, Datensätze in der Zieltabelle zu ändern |
| 726 | In der Importdatei sind mehr Datensätze vorhanden als in der Zieltabelle. Nicht alle Datensätze werden importiert |
| 727 | In der Zieltabelle sind mehr Datensätze vorhanden als in der Importdatei. Nicht alle Datensätze werden aktualisiert |
| 729 | Fehler beim Import. Datensätze konnten nicht importiert werden |
| 730 | Nicht unterstützte Excel-Version. Konvertieren Sie die Datei in Excel 7.0 (Excel 95), Excel 97, 2000, XP oder 2007 und versuchen Sie es erneut. |
| 731 | Die für den Import ausgewählte Datei enthält keine Daten |
| 732 | Diese Datei kann nicht eingefügt werden, weil sie selbst weitere Dateien enthält |
| 733 | Eine Tabelle kann nicht in sich selbst importiert werden |
| 734 | Dieser Dateityp kann nicht als Bild dargestellt werden |
| 735 | Dieser Dateityp kann nicht als Bild dargestellt werden. Er wird eingefügt und dann als Datei dargestellt |

| Fehlernummer | Beschreibung |
|---------------------|--|
| 736 | Zu viele Daten für den Export in dieses Format. Sie werden abgeschnitten |
| 737 | Die Bento-Tabelle, die Sie importieren, fehlt. |
| 800 | Datei konnte auf Datenträger nicht erstellt werden |
| 801 | Temporärdatei konnte auf Systemdatenträger nicht erstellt werden |
| 802 | Datei konnte nicht geöffnet werden. Dieser Fehler kann durch eine oder mehrere der folgenden Möglichkeiten verursacht werden: <ul style="list-style-type: none"> ■ Ungültiger Datenbankname ■ Datei ist in FileMaker Server geschlossen ■ Ungültige Berechtigung |
| 803 | Datei in Einzelbenutzer-Status oder Host nicht vorhanden |
| 804 | Datei konnte in ihrem aktuellen Status nicht mit Nur-Lese-Zugriff geöffnet werden |
| 805 | Datei ist beschädigt; stellen Sie die Datei wieder her |
| 806 | Datei kann mit dieser Version von FileMaker Pro nicht geöffnet werden |
| 807 | Datei ist keine FileMaker Pro-Datei oder ist schwer beschädigt |
| 808 | Datei kann wegen beschädigter Zugriffsrechte nicht geöffnet werden |
| 809 | Datenträger voll |
| 810 | Datenträger fixiert |
| 811 | Temporärdatei kann nicht als FileMaker Pro-Datei geöffnet werden |
| 813 | Fehler bei Datensatz-Synchronisation im Netzwerk |
| 814 | Datei(en) kann (können) nicht geöffnet werden, da die maximale Anzahl an Dateien geöffnet ist |
| 815 | Referenzdatei konnte nicht geöffnet werden |
| 816 | Datei konnte nicht konvertiert werden |
| 817 | Der Bindungsschlüssel der Datei stimmt nicht mit dieser Laufzeitanwendung überein |
| 819 | Lokale Kopie einer remote Datei kann nicht gespeichert werden |
| 820 | Datei wird geschlossen |
| 821 | Host hat die Verbindung getrennt |
| 822 | FMI-Dateien nicht gefunden; fehlende Dateien neu installieren |
| 823 | Datei kann nicht auf Einzelbenutzer gesetzt werden, da Gäste verbunden sind |
| 824 | Datei ist beschädigt oder keine FileMaker-Datei |
| 825 | Datei ist nicht autorisiert, die geschützte Datei zu referenzieren |
| 900 | Allgemeiner Fehler in der Rechtschreibprüfung |
| 901 | Standardwörterbuch nicht installiert |
| 902 | Hilfe-System konnte nicht gestartet werden |
| 903 | Befehl kann nicht in einer gemeinsam genutzten Datei verwendet werden |
| 904 | Befehl kann nur in einer Datei verwendet werden, die von FileMaker Pro Server freigegeben wurde |
| 905 | Kein aktives Feld ausgewählt, Befehl kann nur mit aktivem Feld verwendet werden |
| 906 | Die aktuelle Datei wird nicht bereitgestellt. Der Befehl kann nur verwendet werden, wenn die Datei bereitgestellt wird |
| 920 | Initialisierung der Rechtschreibprüfung nicht möglich |

| Fehlernummer | Beschreibung |
|---------------------|--|
| 921 | Anwenderwörterbuch kann nicht zur Bearbeitung geladen werden |
| 922 | Anwenderwörterbuch existiert nicht |
| 923 | Anwenderwörterbuch ist schreibgeschützt |
| 951 | Ein unerwarteter Fehler ist aufgetreten |
| 954 | Nicht unterstützte XML-Grammatik |
| 955 | Kein Datenbankname |
| 956 | Maximale Anzahl von Datenbanksitzungen überschritten |
| 957 | Widersprüchliche Befehle |
| 958 | Parameter fehlt in Query |
| 959 | Custom Web Publishing ist deaktiviert |
| 1200 | Generischer Rechenfehler |
| 1201 | In dieser Funktion gibt es zu wenige Parameter |
| 1202 | In dieser Funktion gibt es zu viele Parameter |
| 1203 | Unerwartetes Ende der Berechnung |
| 1204 | Es wird eine Zahl, eine Textkonstante, ein Feldname bzw. „(“ erwartet |
| 1205 | Kommentar ist nicht mit „*/“ beendet |
| 1206 | Textkonstante muss mit einem Anführungszeichen enden |
| 1207 | Klammer unvollständig |
| 1208 | Operator fehlt, Funktion nicht gefunden oder „(“ nicht erwartet |
| 1209 | Name (z. B. Feldname oder Layoutname) fehlt |
| 1210 | Plugin-Funktion wurde bereits registriert |
| 1211 | In dieser Funktion ist die Listennutzung nicht zulässig |
| 1212 | Hier wird ein Operator (z. B. +, -, *) erwartet |
| 1213 | Diese Variable wurde bereits in der SetzeVars-Funktion definiert |
| 1214 | MITTELWERT, ANZAHL, ERWEITERN, HOLEWIEDERHOLFELDWERT, MAX, MIN, NBW, STABW, SUMME und HOLEERGEBNISWERT: Ausdruck gefunden, wo nur ein Feld benötigt wird |
| 1215 | Dieser Parameter ist ein ungültiger Statusfunktionsparameter |
| 1216 | Als erstes Argument einer HOLEERGEBNISWERT-Funktion können nur Statistikfelder angegeben werden |
| 1217 | Gruppierfeld ist ungültig |
| 1218 | Zahl kann nicht berechnet werden |
| 1219 | Ein Feld kann nicht in seiner eigenen Formel benutzt werden |
| 1220 | Feldtyp muss normal oder berechnet sein |
| 1221 | Datentyp muss Zahl, Datum, Zeit oder Zeitstempel sein |
| 1222 | Formel kann nicht gespeichert werden |
| 1223 | Die angegebene Funktion ist noch nicht implementiert |
| 1224 | Die angegebene Funktion existiert nicht |

| Fehlernummer | Beschreibung |
|--------------|--|
| 1225 | Die angegebene Funktion wird in diesem Kontext nicht unterstützt |
| 1400 | ODBC-Client-Treiberinitialisierung fehlgeschlagen. Stellen Sie sicher, dass die ODBC-Client-Treiber richtig installiert sind |
| 1401 | Umgebung konnte nicht zugeteilt werden (ODBC) |
| 1402 | Umgebung konnte nicht freigegeben werden (ODBC) |
| 1403 | Verbindung trennen fehlerhaft (ODBC) |
| 1404 | Verbindung konnte nicht zugeteilt werden (ODBC) |
| 1405 | Verbindung konnte nicht freigegeben werden (ODBC) |
| 1406 | Überprüfung von SQL API fehlgeschlagen (ODBC) |
| 1407 | Anweisung konnte nicht zugeteilt werden (ODBC) |
| 1408 | Erweiterter Fehler (ODBC) |
| 1409 | Erweiterter Fehler (ODBC) |
| 1410 | Erweiterter Fehler (ODBC) |
| 1411 | Erweiterter Fehler (ODBC) |
| 1412 | Erweiterter Fehler (ODBC) |
| 1413 | Erweiterter Fehler (ODBC) |
| 1450 | Aktion erfordert PHP-Berechtigungserweiterung |
| 1451 | Aktion erfordert, dass die aktuelle Datei remote ist |
| 1501 | SMTP-Authentifizierung fehlgeschlagen |
| 1502 | Verbindung vom SMTP-Server abgelehnt |
| 1503 | SSL-Fehler |
| 1504 | Der SMTP-Server erfordert eine verschlüsselte Verbindung |
| 1505 | Die angegebene Authentifizierung wird vom SMTP-Server nicht unterstützt |
| 1506 | E-Mail(s) konnte(n) nicht erfolgreich versendet werden |
| 1507 | Anmeldung am SMTP-Server nicht möglich |

Fehlercodenummern für PHP-Komponenten

Das FileMaker API for PHP nutzt eine Reihe von PHP-Komponenten. Diese PHP-Komponenten können weitere Fehlercodes zurückgeben, die hier nicht aufgelistet werden.

Wenn beispielsweise Web Publishing Core- oder FileMaker Server-Dienste nicht laufen, kann der cURL-Modulfehler „CURLE_GOT_NOTHING (52)“ zurückgegeben werden.

Weitere Informationen zu PHP-Fehlercodes finden Sie auf der PHP-Website <http://php.net>.

Index

A

add(), Methode 45
addSortRule(), Methode 43
Admin Console 18, 19
Aktivieren von Instant Web Publishing in Datenbank 19
Aus Werteliste, Überprüfung 49
Ausführen von Suchabfragen 43
Ausschnitte 40
Ausschnitteinstellungen, Dialogfeld 47

B

Bearbeiten von Datensätzen 37
Befehl „Add“ 36
Beispiele zu FileMaker API for PHP 34
Benutzernamen
 Für Custom Web Publishing definieren 19
Berechtigung, Zuweisen für Custom Web Publishing 19
Bezugsmengenobjekt 40

C

clearSortRules(), Methode 43
Client URL Library 16
commit(), Methode 36
Compound Find
 Befehl 45
 Beispiel 46
createRecord(), Methode 36
cURL 16
cURL-Modulfehler 63
Custom Web Publishing
 Definition 9
 Erweitertes Zugriffsrecht für 19
 In Datenbank aktivieren 19
 In Web Publishing Engine aktivieren 20
 Mit PHP 11
 Mit XML 11
 Mit XSLT 11
 Scripts 24
 Scripts verwenden 23
 Übersicht 9
 Zugriff von IP-Adressen in Web-Server beschränken 20

D

date, Feld 49
Datenbanken, veröffentlichte schützen 20
Datenbankobjekt 35
Datenbanksitzungen, Persistenz 19, 21
Datensätze 36
Datensatzobjekt 36
Datumsdarstellung 47
delete(), Methode 37, 41
Delete, Befehl 37
Dokumentation 7
Duplicate, Befehl 36
Duplizieren von Datensätzen 36
Dynamische IP-Adresse 16

E

Edit, Befehl 37
Eindeutiger Wert, Überprüfung 49
Einstellen von Websites 53
Elektronische Dokumentation 7
Ergebnismenge 47
Erneut anmelden, Script 21
Erstellung eines Datensatzes 36
Erweitertes Zugriffsrecht für Custom Web Publishing 19
Externe SQL-Datenquelle 19

F

Fehler
 Behandlung 51
 Datenbankfehlercodes 57
 Protokolldateien für Web-Server 54
Fehlerbehandlung 51
Fehlerbehebung
 Custom Web Publishing-Websites 53
Fehlerbehebung für Websites 56
Feld
 Datum 49
 four-digit year 48
 maximum number of characters 48
 nicht leer 48
 numeric only 48
 time of day 49
 Zeit 49

- Zeitstempel 49
- FileMaker API for PHP 11
 - Beispiele 34
 - Definition 11
 - Lehrgang 34
 - Manuelle Installation 17
 - Referenz 33
- FileMaker Server
 - Dokumentation 7
 - Installieren 7
- FileMaker Server Admin
 - Siehe Admin-Konsole
- FileMaker, Klasse 34
- FileMaker, Objekte der Klasse 35
 - Bezugsmenge 40
 - Datenbank 35
 - Datensatz 36
- FileMaker-Befehlsobjekte
 - Bearbeiten 37
 - Compound Find, Befehl 45
 - Duplizieren 36
 - Find All, Befehl 44
 - Find Any, Befehl 44
 - Find, Befehl 43, 44
 - Hinzufügen 36
 - Löschen 37
- Find All, Befehl 44
- Find Any, Befehl 44
- Find, Befehl 44
- four-digit year, Feld 48

G

- Gastkonto
 - Aktivieren 21
 - Deaktivieren 21
- Generieren von PHP-Code 31
- getDatabase(), Methode 39
- getErrors(), Methode 52
- getFetchCount(), Methode 47
- getField(), Methode 47
- getFieldAsTimestamp(), Methode 47
- getFields(), Methode 39, 47
- getFoundSetCount(), Methode 47
- getLayout(), Methode 39
- getMessage(), Methode 52
- getName(), Methode 39, 40
- getRange(), Methode 43
- getRecords(), Methode 47
- getRelatedSet(), Methode 40

- getRelatedSets(), Methode 40
- getValueListsTwoFields(), Methode 42
- getValueListTwoFields(), Methode 42
- GIF-Dateien, im Web veröffentlichen 22

H

- Herstellen einer Verbindung
 - mit einem FileMaker Server 35
 - mit einer FileMaker-Datenbank 35

I

- Im Bereich, Überprüfung 49
- Installation von FileMaker API for PHP 17
- Installationsdokumentation 7
- Instant Web Publishing
 - Definition 9
 - Dokumentation 7
- isError(), Methode 51
- isValidationError(), Methode 50

J

- JDBC-Dokumentation 7
- JPEG-Dateien, im Web veröffentlichen 22

K

- Kodierung Latin-1 28
- Kodierung UTF-8 28
- Konten und Zugriffsrechte
 - Für Custom Web Publishing aktivieren 19
 - Gastkonto 21
 - Scripts 23

L

- Layouts 39
- Lehrgang zu FileMaker API for PHP 34
- listFields(), Methode 39
- listLayouts(), Methode 39
- listRelatedSets(), Methode 39, 40
- listScripts(), Methode 38
- listValueLists(), Methode 39, 42
- Löschen von Datensätzen 37

M

- Mac OS X Server Admin 16
- Manuelle Installation von FileMaker API for PHP 17

maximum number of characters, Feld 48

Medienfelder

Inhalt veröffentlichen 21

Wie Web-Benutzer auf Daten zugreifen 22

Methoden

add() 45
 addSortRule() 43
 clearSortRules() 43
 commit() 36
 CreateRecord() 36
 delete() 37, 41
 getDatabase() 39
 getErrors() 52
 getFetchCount() 47
 getField() 47
 getFieldAsTimestamp() 47
 getFields() 39, 47
 getFoundSetCount() 47
 getLayout() 39
 getMessage() 52
 getName() 39, 40
 getRange() 43
 getRecords() 47
 getRelatedSet() 40
 getRelatedSets() 40
 getValueListsTwoFields() 42
 getValueListTwoFields() 42
 isError() 51
 isValidationError() 50
 listFields() 39
 listLayouts() 39
 listRelatedSets() 39, 40
 listScripts() 38
 listValueLists() 39, 42
 newAddCommand() 36
 newCompoundFindCommand() 45
 newDeleteCommand() 37
 newDuplicateCommand() 36
 newEditCommand() 37
 newFindAllCommand() 44
 newFindAnyCommand() 44
 newFindCommand() 44
 newFindRequest() 45
 newPerformScriptCommand() 38
 newRelatedRecord() 41
 numErrors() 50, 51
 setLogicalOperator() 43
 setOmit() 45
 setPreCommandScript() 38, 44
 setPreSortScript() 38, 44
 setProperty() 35
 setRange() 43

setRelatedSetsFilters() 47

setResultsLayout() 39

setScript() 38, 44

validate() 49

N

newAddCommand(), Methode 36
 newCompoundFindCommand(), Methode 45
 newDeleteCommand(), Methode 37
 newDuplicateCommand(), Methode 36
 newEditCommand(), Methode 37
 newFindAllCommand(), Methode 44
 newFindAnyCommand(), Methode 44
 newFindCommand(), Methode 44
 newFindRequest(), Methode 45
 newPerformScriptCommand(), Methode 38
 newRelatedRecord(), Methode 41
 nicht leeres Feld 48
 numErrors(), Methode 50, 51
 Nummern für
 Datenbankfehlercodes 57
 nur numerisches Feld 48

O

ODBC-Beschränkungen 19
 ODBC-Dokumentation 7
 Online-Dokumentation 7

P

Passwort ändern (Script) 21
 Passwörter
 Für Custom Web Publishing definieren 19
 Kein Anmeldepasswort 21
 Passwort ändern (Script) 21
 Persistente Datenbanksitzungen 19, 21
 PHP
 Fehler 63
 Fehlerbehebung 56
 In Datenbank aktivieren 19
 Unterstützte Version 17
 Vorteile 13
 Websites testen 53
 Zusammenfassung der Schritte für Publishing 27
 PHP 5 16
 PHP API für Custom Web Publishing 11
 PHP-Publishing testen 56
 PHP-Site-Assistent
 Starten 29

Verwendung von generiertem Code 31

Verwendung vorbereiten 29

Programmprotokoll 55

Protokolldateien 53, 55

Beschreibung 54

web_server_module_log.txt 55

Web-Server-Zugriff 54

Q

QuickTime-Filme, im Web veröffentlichen 22

R

Referenzinformationen 33

S

SAT

Siehe Admin-Konsole

Schützen veröffentlichter Datenbanken 20

Scripts 37

Erneut anmelden 21

In Custom Web Publishing 23

Konten und Zugriffsrechte 23

Passwort ändern 21

Tipps und Überlegungen 23

Trigger 25

Server Admin-Tool

Weitere Informationen erhalten Sie unter Mac OS X Server Admin.

Serveranforderungen 16

setLogicalOperator(), Methode 43

setOmit(), Methode 45

setPreCommandScript(), Methode 38, 44

setPreSortScript(), Methode 38, 44

setProperty(), Methode 35

setRange(), Methode 43

setRelatedSetsFilters(), Methode 47

setResultsLayout(), Methode 39

setScript(), Methode 38, 44

Sicherheit

Dokumentation 11

Konten und Passwörter 20

Richtlinien zum Schützen veröffentlichter Datenbanken 20

Zugriff von IP-Adressen beschränken 20

Site-Assistent *siehe* PHP-Site-Assistent

SSL (Secure Sockets Layer)-Verschlüsselung 20

Statische IP-Adresse 16

Statisches Veröffentlichen, Definition 9

Suchbefehlsobjekte 43

T

Technologietests 56

Testen von Websites 53

time of day, Feld 49

time, Feld 49

timestamp, Feld 47, 49

Trigger 25

U

Überprüfung 48

Befehle 48

Datensätze 50

Datum 49

Felder 50

four-digit year 48

maximum number of characters 48

nicht leer 48

numeric only 48

time of day 49

Zeit 49

Zeitstempel 49

Überprüfung durch Berechnung 49

Übersicht

Custom Web Publishing 9

PHP-Publishing 27

Überwachen von Websites 54

Unicode 28

Unix-Zeitstempel 47

V

validate(), Methode 49

Verarbeitung einer Ergebnismenge 47

Verarbeitung einer Web Publishing Engine-Anforderung 10

Veröffentlichen im Web

Datenbankfehlercodes 57

Medienfeldobjekte 21

QuickTime-Filme 22

Schützen von Datenbanken 20

Verwendung von PHP 27

Verwenden

Ausschnitte 40

Datensätze 36

Layouts 39

Scripts 37

Wertelisten 41

Vorüberprüfung 48

- Befehle 48
- Datensätze 50
- Datum 49
- Felder 50
- four-digit year 48
- maximum number of characters 48
- nicht leer 48
- numeric only 48
- time of day 49
- Zeit 49
- Zeitstempel 49
- Vorhandener Wert, Überprüfung 49
- Vorteile von XML und XSLT 13

W

- Web Publishing Core
 - Interne Zugriffsprotokolle 55
- Web Publishing Engine
 - Anforderungsverarbeitung 10
 - Beschreibung 10
 - Erzeugte Fehlercodes 57
 - Programmprotokoll 55
- web_server_module_log.txt (Protokolldatei) 55
- Web-Benutzer
 - Medienfelddaten verwenden 22
- Webbrowser
 - Ausgabe empfangen 10
- Web-Ordner, Medienfeldobjekte kopieren 22
- Web-Server
 - Protokolldateien 54
- Websites
 - Einstellen 53
 - Fehlerbehebung 56
 - FileMaker Support-Seiten 7
 - Testen 53
 - Überwachen 54
- Wertelisten 41
- wpc_access_log.txt (Datei) 55

X

- XML Custom Web Publishing 11
- XSLT Custom Web Publishing 11

Z

- Zugriffsprotokolldateien für Web-Server,
 - Beschreibung 54
- Zugriffsrechte 21

