



Bernhard Schulz

(Jg. 1981) ist Inhaber und Geschäftsführer der schubec GmbH (www.schubec.com). Die Firma bietet Dienstleistungen und Produkte rund um FileMaker wie Plugins, Schulungen, Consulting, PHP und Web Hosting.

bernhard.schulz@schubec.com

Web-Publishing mit FileMaker

Wie kommen meine Daten ins Internet?

Sie möchten Inhalte aus Ihrer FileMaker Datenbank im Internet veröffentlichen oder wollen Anfragen von Kunden entgegennehmen und die Daten in FileMaker speichern? Dann heiÙe ich Sie herzlich willkommen in der Welt des Web-Publishing mit FileMaker!

Vorab kann ich diejenigen beruhigen, die sich bisher noch nicht an das Thema herangetraut haben: Web-Publishing mit FileMaker funktioniert problemlos! Die Frage ist, welchen technischen Weg Sie einschlagen, um Ihr Ziel zu erreichen, denn Möglichkeiten gibt es viele. In diesem Beitrag stelle ich die unterschiedlichen Varianten vor, verweise auf einige damit umgesetzte Projekte und erläutere Ihnen die jeweiligen Vor- und Nachteile – unter anderem anhand einer imaginären Ferienhausvermietung auf Mallorca.

Web-Publishing Möglichkeiten

1. Export von statischen HTML-Daten aus FileMaker
2. FileMaker WebDirect
3. FileMaker Custom Web Publishing
4. SQL-Datenbank mit FileMaker ESS-Zugriff
5. Beliebige Datenhaltung mit FileMaker REST-Zugriff

Beispiel Ferienhausvermietung

Nehmen wir an, Sie vermieten zehn Ferienhäuser auf Mallorca. Zwei davon sind Ihre eigenen, die restlichen acht gehören drei Bekannten, in deren Namen Sie die Häuser für eine kleine Provision vermitteln. In FileMaker haben Sie alle zehn Häuser ansprechend beschrieben, inklusive Fotos und den Daten, wann die Häuser an wen vermietet sind oder von den Besitzern selbst benötigt werden.



1. Export von statischen HTML-Daten aus FileMaker

Bei der „langweiligsten“ Variante bauen Sie – oder die von Ihnen beauftragte Person – in FileMaker mithilfe von Scripts und Formeln ein paar nette HTML-Codeblöcke zusammen. Anschließend wird per Tastendruck ein Script aufgerufen, mit dem die Daten als HTML-Datei und als Bilder exportiert und möglicherweise per FileMaker Plugin sofort auf den Webserver hochgeladen werden.

Diese Variante hat den **Vorteil**, dass sie kaum Anforderungen an den Webserver stellt – jeder Webserver kann statische HTML-Dateien und Bilder veröffentlichen. Zudem ist das Ganze rasend schnell und im Normalfall gut für Suchmaschinen lesbar. Sie benötigen keinen FileMaker Server, es reicht der Client.

Ein **Nachteil** ist, dass Sie die HTML-Daten in FileMaker „zusammenbauen“ oder – falls Sie sie nicht direkt als HTML sondern z. B. als JSON oder XML exportieren wollen – mit einer weiteren Software in HTML-Dateien konvertieren müssen. Es werden also mindestens Programmierkenntnisse in HTML benötigt.

Beispiel Ferienhausvermietung

Der größte **Nachteil** ist, dass die Endkunden keine gute Möglichkeit haben, das Ferienhaus direkt online zu mieten, da

keine Verbindung von der Webseite zu FileMaker besteht. Sie können höchstens eine Telefonnummer oder ein E-Mail-Formular auf der Webseite hinterlegen und die Kunden buchen dann darüber.

Praxisbeispiel

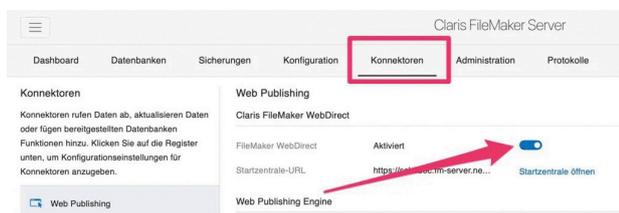
Dieses Verfahren habe ich bei meinem Projekt **Salzburg Sehenswürdigkeiten** gewählt. Hier werden JSON-Daten aus FileMaker exportiert, mit einem Javaprogramm in HTML umgewandelt und auf den Webserver geladen.

<https://salzburg-sehenswuerdigkeiten.at/>

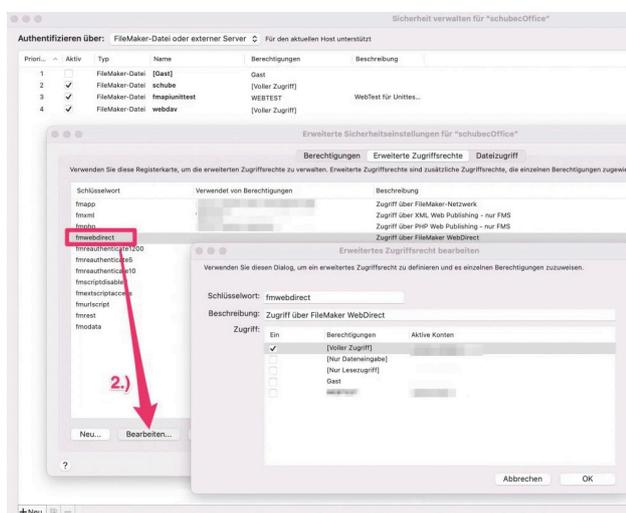
2. FileMaker WebDirect – Web-Publishing ohne Programmierung

FileMaker WebDirect ist für viele Einsatzszenarien eine ganz tolle Möglichkeit, da man die FileMaker Datenbank samt Layouts und der meisten Funktionen problemlos im Internet veröffentlichen kann, ohne speziell etwas dafür zu programmieren: Wenn Sie FileMaker Layouts und Scripts beherrschen, dann können Sie auch WebDirect!

Für die Verwendung dieser Technologie wird zwingend ein FileMaker Server benötigt, der über das Internet zugänglich ist. Aktivieren Sie WebDirect zunächst am FileMaker Server im Menüpunkt „Konnektoren“.



Danach sollten Sie in der zu veröffentlichen Datenbank individuelle Berechtigungen setzen, denn Sie wollen vermutlich nicht die gesamte Datenbank für anonyme Gastbenutzer mit Lese- und Schreibrechten im Internet veröffentlichen ...



Wie bereits erwähnt ist der **Vorteil** von WebDirect, dass Sie keine Programmierkenntnisse in HTML/JavaScript/CSS benötigen, sondern mit Ihrem vorhandenen FileMaker Wissen Datenbanken im Internet veröffentlichen können.

Aber es gibt auch gravierende **Nachteile!** So ist WebDirect im Normalfall sehr kostenintensiv: **Clariss** verlangt pro Benutzer eine FileMaker Lizenz oder berechnet sogenannte „Connections“. Geht man von zehn gleichzeitigen WebDirect-Benutzern aus, benötigt man eine Lizenz für zehn Concurrent Connections, die dreimal so teuer sind wie „normale“ FileMaker Lizenzen. Und zehn Webseitenbenutzer sind nicht viel: Wenn Sie mit Ihrer Ferienhausvermietung erfolgreich sind, befinden sich zu Stoßzeiten möglicherweise 100 oder sogar 1000 Interessenten gleichzeitig auf Ihrer Webseite.

Wollen Sie sich nicht auf zehn, 20 oder 50 gleichzeitige Benutzer festlegen, benötigen Sie eine unlimitierte Lizenz. Das geht dann in Richtung FileMaker Firmenlizenz. Genaue Preise erfragen Sie beim Lizenzhändler Ihrer Wahl.

Außerdem ist FileMaker WebDirect sehr ressourcenintensiv. Im Prinzip wird pro Benutzer am Server ein interner FileMaker Client (ohne Oberfläche) gestartet. Wenn zehn Benutzer gleichzeitig arbeiten, benötigt man schon einen gut ausgestatteten Server.

Der dritte große Nachteil ist, dass die WebDirect-Seiten nicht von Suchmaschinen gelesen werden können. Das heißt, auch wenn Sie das tollste Ferienhausangebot mit den besten Preisen haben, wird Sie niemand über **Google** finden, da die WebDirect-Seiten für Suchmaschinen unsichtbar sind. Wenn Sie also per Suchmaschine gefunden werden wollen, ist WebDirect tabu!

Beispiel Ferienhausvermietung

Für unsere Ferienhausvermietung wäre WebDirect optimal, wenn Ihre Hausbesitzer-Freunde per Webbrowser ihre eigenen Häuser selbst als verfügbar oder nicht verfügbar melden, Daten zu zukünftigen Urlaubern einsehen oder Abrechnungen anfordern möchten. Da kämen Sie mit der kleinsten verfügbaren Lizenz (5 Benutzer inkl. Servern usw.), aus.

FileMaker WebDirect ist eine ganz tolle Sache, die ich in vielen Kundenprojekten verwende. Allerdings empfiehlt sich der Einsatz nur für eine kleine, definierte Gruppe, die per Webbrowser auf FileMaker zugreift. Für „normale Webseiten“, die im Internet veröffentlicht werden sollen, ist diese Technologie vollkommen ungeeignet. Praktisch ist, dass man WebDirect mit anderen Web-Publishing-Varianten mischen kann, sodass man für administrative Aufgaben WebDirect zur Verfügung stellt, die eigene Webseite aber mit einer der anderen Möglichkeiten veröffentlicht.

Praxisbeispiel

Hier kann ich kein Beispiel verlinken, da WebDirect bei mir nur für administrative und mit Passwort geschützte Zugänge in Frage kommt und nicht für öffentliche Internetseiten.

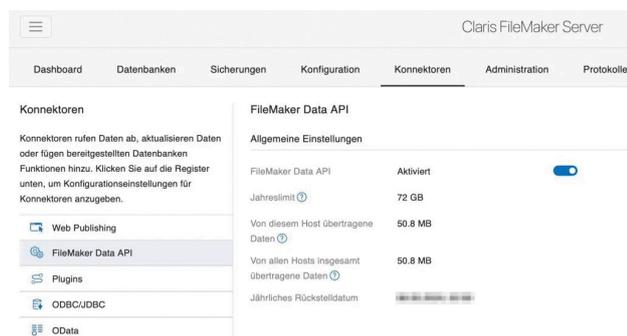
3. FileMaker Custom Web Publishing

Beim Custom Web Publishing dient FileMaker im Prinzip nur als Datenbank. Der gesamte Webauftritt inklusive der Abläufe wird in einer Programmiersprache wie PHP, Java, C#, Ruby, Python, JavaScript/Node o.ä. neu programmiert.

Es werden keine FileMaker Layouts genutzt, sondern die gesamte Oberfläche wird als HTML/CSS/JavaScript aufgesetzt. Das ist einerseits viel Arbeit, andererseits hat man natürlich ganz andere (bessere) Möglichkeiten, Layouts zu gestalten. Die gesamte Logik, mit der FileMaker Scripts agieren, wird weitgehend ignoriert, und sämtliche für den Webauftritt benötigten Funktionen werden neu programmiert. Es ist zwar möglich FileMaker Scripts aufzurufen und so gewisse Abläufe darüber zu steuern, aber die native Programmierung in PHP/Java/etc. ist um ein Vielfaches schneller.

In der Praxis erstelle ich bei solchen Projekten 96 % des Codes in PHP/Java neu und rufe nur vereinzelt Scripts in FileMaker auf, beispielsweise um per E-Mail Anmeldebestätigungen mit PDF-Anhang zu versenden. So kann der FileMaker Programmierer sich beispielsweise um die Formatierung der PDF-Dokumente kümmern.

Der Zugriff auf die FileMaker Datenbank erfolgt per Data API, die am FileMaker Server aktiviert werden muss.



Prinzipiell lässt sich *Claris* auch die FileMaker Data API bezahlen, doch man bekommt pro Benutzer und Jahr 24 GB Datenvolumen zur freien Verfügung. Bei einem Standardvertrag mit fünf Benutzern sind das 120 GB jährlich und das Übertragen von Medienfeldern wird nicht angerechnet. Das reicht für 99,9% aller Anwendungen locker aus.

Vorteil des Custom Web Publishing ist, dass alle Daten live aus FileMaker ausgelesen oder dahin zurückgeschrieben werden können.

Die Webseitenlayouts müssen neu gestaltet werden, sodass man hübsche, ansprechende und responsive Webseiten erstellen kann, die bei entsprechender Programmierung von Suchmaschinen gelesen werden können.

Die Nutzung der Data API selbst ist eher kompliziert, sodass fast niemand diese direkt nutzt. In der Regel verwendet man eine „Wrapper-Bibliothek“, die den Zugriff mithilfe einer geeigneten Programmiersprache stark vereinfacht. Die momentan verfügbaren Libraries finden Sie unter

https://support.claris.com/s/article/data-api---admin-api-packages-wrappers-for-fm-17-x?language=en_US

Data API

Language	Package	Author	URL
C#	fmdata	Nate Bross	GitHub - fuzzerd/fmdata: A simple .NET Standard client for the FileMaker Data API.
FileMaker Scripting	fmPDA	Mark DeNyse	GitHub - driftwoodinteractive/fmPDA: A replacement class for the FileMaker API For PHP using the FileMaker Data. fmPDA p...
Java	Android-Fm	Jose E. Lopez	GitHub - joselopezrosario/filemaker-data-api: An Android library to seamlessly interact with databases hosted on FileMak...
Javascript	FileMaker-Data-API-Demo	Steve Winter	GitHub - matatrosolutions/filemaker-data-api-demo: Demo JavaScript used for FileMaker DevCon 2018 session "Introduction..."
Javascript	FMS-API-Client	Lui de la Parra	GitHub - Luidog/fms-api-client: A FileMaker Server Data API client designed to allow interaction with a FileMaker applic...
Javascript	node-red-contrib-filemaker	Lui de la Parra	GitHub - Luidog/node-red-contrib-filemaker: A Node Red FileMaker module. This module uses the FileMaker Data API to conn...
.NET	fmdata	Nate Bross	GitHub - fuzzerd/fmdata: A simple .NET Standard client for the FileMaker Data API.
.NET	FMDotNet_DataAPI	Wim Decorte	GitHub - wimdecorte/FMDotNet_DataAPI: .NET wrapper (.NET Standard 2.0) around the FileMaker Server REST Data API.
PHP	FMDDataAPI	Masayuki Nii	GitHub - msyk/FMDDataAPI: FMDDataAPI is a class developed by PHP to access FileMaker database with FileMaker Data API.
PHP	fmPDA	Mark DeNyse	GitHub - driftwoodinteractive/fmPDA: A replacement class for the FileMaker API For PHP using the FileMaker Data. fmPDA p...
PHP	fmREST	Ken d'Oronzio	GitHub - kdoronzio/fmREST.php: PHP class file to support FileMaker Server 17 Data API (REST engine)
PHP	fmRESTor	24u	GitHub - 24u/fmRESTor: Leverage the FileMaker® 17 Data API with ease!
PHP/Javascript	INTER-Mediator	INTER-Mediator Directive Committee	INTER-Mediator Release INTER-Mediator 5.8 released · INTER-Mediator/INTER-Mediator · GitHub
PostMan	FMS Data API 17	Wim Decorte	Dropbox - FMS Data API 17.postman_collection.json
Python	fmrest	Eric Luce	GitHub - eluce2/fmrest: a Python library for interacting with the FileMaker REST APIs
Python	python-fmrest	David Hamann	GitHub - davidhamann/python-fmrest: Python wrapper around the FileMaker Data API
Ruby	fmrest-ruby	pcarbajal	https://github.com/beezwax/fmrest-ruby
Swift	SwiftFM	Brian Hamm	GitHub - starsite/SwiftFM: A service class for Swift to work with the FileMaker Data API

Ich nutze für meine Projekte meistens die von mir programmierte Java-Variante, die Sie kostenlos von **GitHub**¹ laden können. In meinen Schulungen zu FileMaker und PHP nutzen wir die Bibliothek von **INTER-Mediator**².

In der Praxis findet man genug Entwickler, die zum Beispiel mit PHP eine individuelle Webseite gestalten können. Obwohl die meisten von ihnen keine Erfahrung mit FileMaker haben – sie nutzen zu 99 % MySQL – sollte ein Zugriff auf FileMaker für einen guten PHP-Entwickler mithilfe der INTER-Mediator-Bibliothek kein großes Problem sein.

Der Zugriff auf die FileMaker Data API ist prinzipiell schnell, aber auf eine MySQL-Datenbank kann man 100- bis 1000-mal schneller zugreifen. Daher muss man Programmierer, die erstmals mit der FileMaker Data API zu tun haben, darauf hinweisen, mit den Abfragen sorgsam umzugehen und Daten zu cachen, damit das System performant bleibt.

Beispiel Ferienhausvermietung

Für unsere Ferienhausvermietung lässt sich mithilfe von Custom Web Publishing mit PHP eine wunderschöne, voll responsive Webseite nach neuestem Stand der Technik programmieren. Die Urlaubsgäste können online ein Haus

buchen und per Kreditkarte bezahlen. Alle Daten werden direkt in FileMaker übertragen.

Durch Suchmaschinenoptimierung landen wir auf Position eins bei **Google** und haben viele Anfragen. Wir implementieren daher ein vernünftiges Caching und freuen uns. Da sich kaum ein PHP-Entwickler mit dem Zugriff per FileMaker Data API auskennt, hat sich die Suche nach einer geeigneten Person etwas schwierig gestaltet, aber letztendlich haben wir für unser Projekt jemanden gefunden.

Allerdings trauen wir uns kaum, den FileMaker Server neu zu starten oder Updates einzuspielen, denn in dieser Zeit kann kein Kunde ein Ferienhaus buchen, sondern erhält nur eine Fehlermeldung.

Praxisbeispiele

Ein Projekt, das mit Custom Web Publishing umgesetzt wurde (inkl. Caching), ist die Webseite der **Gemeindebibliothek Dinhard** unter

<https://dinhard.biblioweb.ch>

Ein Webshop-Beispiel finden Sie unter

<https://bingooo.shop/immobingooo/>

4. SQL-Datenbank mit FileMaker ESS-Zugriff

Die nächste Option ist mit der Custom Web Publishing Variante verwandt – auch hier werden die HTML-Seite und die Businesslogik in einer Programmiersprache wie PHP programmiert.

Die Daten kommen aber nicht direkt aus FileMaker, sondern werden aus einer SQL-Datenbank wie MySQL, MariaDB, Oracle SQL, MS SQL oder PostgreSQL gelesen bzw. dahin zurückgeschrieben.

FileMaker wiederum verbindet man per ESS-Technologie über ODBC mit der SQL-Datenbank. Die SQL-Tabellen erscheinen wie Tabellen einer externen Datei und man kann darin Daten lesen, schreiben, importieren, exportieren etc. – mit kleinen Einschränkungen, zum Beispiel bei Medienfeldern. Eine Anleitung zur Einrichtung von ESS finden Sie in meinem Blog³.

Vorteile

- Es gibt viele PHP/MySQL-Entwickler, sodass man leicht Dienstleister findet.
- Geschwindigkeit – der Zugriff auf eine SQL-Datenbank ist schnell.
- Keine/kaum Lizenzkosten – PHP und MariaDB lassen sich kostenlos nutzen.
- Auch an FileMaker sind keine Gebühren zu bezahlen.
- FileMaker Server unter Windows und Linux können kostenlose ODBC-Treiber nutzen.
- FileMaker Server unter macOS benötigen kostenpflichtige ODBC-Treiber von *Actual Technologies* (ca. 300 €).
- FileMaker Server muss nicht per Internet erreichbar sein – diese Aufgabe übernimmt der SQL Server.

Nachteile

- Die Daten müssen irgendwie von FileMaker in die SQL-Datenbank kommen und umgekehrt.
- Sie müssen Import-/Export-Skripts programmieren und regelmäßig ausführen.
- Die meisten Internetprovider mit den typischen 5-Euro-pro-Monat-Paketen bieten zwar PHP und MySQL, erlauben aber keinen Zugriff per ODBC. Hier muss ein passender Provider ausgewählt werden.
- Die Konfiguration der ODBC-Verbindung ist unter Umständen nicht ganz trivial (für Linux und macOS habe ich es in meinem Blog beschrieben⁴).
- Unter Umständen müssen Sie Ihre Firewall konfigurieren, um den ODBC-Zugriff zu erlauben.
- Die ODBC-Verbindung muss mittels SSL-Zertifikat gesichert werden!

Das Rad nicht neu erfinden

Dies ist ein wichtiges Prinzip, das in der Praxis oft eine Rolle spielt. Es gibt hunderttausende OpenSource-Projekte, die alle möglichen Funktionen anbieten, beispielsweise das beliebte CMS **WordPress** mit seinen Millionen von Plugins.

Sehr viele dieser Projekte speichern die Daten in einer **MySQL/MariaDB**-Datenbank, die man meistens relativ einfach per ODBC auslesen kann. Daher sollten Sie, bevor Sie selbst etwas entwickeln oder beauftragen, am besten mal bei **GitHub**⁵ vorbeischaun und prüfen, ob es nicht schon ein fertiges Projekt gibt, das Sie nutzen können.

Achtung Updates!

Angenommen, Sie installieren einen OpenSource-Webshop, wie beispielsweise *PrestaShop* und greifen direkt auf die SQL-Datenbank zu, dann werden Sie damit eine Zeitlang zufrieden arbeiten können. Wenn Sie aber eines Tages ein Update installieren, das die interne Datenbankstruktur ändert, funktionieren Ihre Datenbankzugriffe per ODBC nicht mehr und Sie müssen nacharbeiten.

Die Anbieter solcher Systeme gehen im Allgemeinen einfach nicht davon aus, dass jemand direkt auf die SQL-Daten zugreift und werden Sie in der Regel nicht über solche Datenbank-Updates in den Release Notes informieren. Hier ist also Vorsicht geboten!

Das heißt: Wenn Sie per FileMaker/ESS auf eine selbst programmierte SQL-Datenbank zugreifen, gibt es kein Problem, denn dann wissen Sie ja, ob Sie etwas ändern und was von der Änderung betroffen ist. Wenn Sie aber die bestehende Datenbank eines OpenSource-Projektes „anzapfen“, sollten Sie früher oder später mit Turbulenzen rechnen!

Beispiel Ferienhausvermietung

Für unsere Ferienhausvermietung nutzen wir die Webseite aus dem dritten Szenario mit FileMaker Custom Web Publishing weiter, tauschen jedoch den Datenzugriffsteil aus, indem wir die FileMaker Data API entfernen und auf eine **MariaDB**-SQL-Datenbank setzen.

Die Webseite ist nun noch schneller als zuvor. Da wir keine exotischen Technologien (hier: FileMaker Data API) mehr einsetzen, sondern Standard PHP/MySQL bzw. **MariaDB** nutzen, finden wir leicht Entwickler, denn das ist deren täglich Brot. Den Datenaustausch können wir mit FileMaker Bordmitteln selbst programmieren.

FileMaker Server können wir rebooten, wann immer wir wollen, denn die Webseite greift auf den SQL-Server beim Provider zu und ist immer online – wir sind glücklich!

Praxisbeispiele

- Buchen Sie Tickets bei **Princess for one day**: zum Einsatz kommen in diesem Projekt **MariaDB** und FileMaker/ESS.
- Buchen Sie Tickets bei **Konzert-Wien.de**: zum Einsatz kommen hier **MariaDB** und FileMaker/ESS.
- Lernen Sie Klavier bei **Austrian Master Classes**. In dieser Lösung werden FileMaker und ESS verwendet.
- Als Schüler im Bundesland Salzburg können Sie Ihren Schülerschein und das Jahres-Busticket hier bestellen: <https://bestellung.s-pass.at/>
Ich glaube, Sie können erraten, wie dieses Projekt umgesetzt wurde – mit FileMaker und ESS.

Sie haben hier unterschiedliche Webseiten, Layouts und Designs kennengelernt und gesehen: Alles ist möglich. Dass im Hintergrund alles letztlich über FileMaker läuft, erkennen Sie als Endanwender nie.

5. Beliebige Datenhaltung mit FileMaker REST-Zugriff

Kommen wir nun zur letzten Variante: Wieder haben wir eine Webseite programmiert, die Daten werden gespeichert. Sei es in einer SQL-Datenbank, wie im vierten Szenario, sei es in einer No-SQL-Datenbank wie **MongoDB** oder in einfachen Textdateien – das „Wie“ spielt keine Rolle. Wichtig ist, dass die Webseite eine REST-API zur Verfügung stellt.

Mittels dieser REST-API können Daten zwischen der Webseite und einem Fremdsystem – in unserem Fall FileMaker – sowohl lesend als auch schreibend ausgetauscht werden. In FileMaker lässt sich die REST-APIs mit dem Befehl „Aus URL einfügen...“ ganz ohne Plugins nutzen. Wie so etwas funktioniert, habe ich unter „cURL-Beispiele für FileMaker anpassen“⁶ beschrieben.

Vorteil ist, dass die meisten OpenSource-Projekte und viele kommerzielle Anbieter für ihre Dienste REST-APIs zur Verfügung stellen und diese meistens gut dokumentiert sind. Die APIs sind dafür da, dass sie von Fremdsystemen genutzt werden und bleiben bei Updates meistens stabil – das heißt,

es kommen möglicherweise Funktionen dazu, aber es werden keine bestehenden Funktionen geändert. Bei dem vierten Szenario, in dem direkt auf die SQL-Datenbank zugegriffen wird, ist das bei Updates nicht immer gewährleistet.

The screenshot shows the 'Web service reference' page from the PrestaShop Developer Conference. It includes a search bar, a navigation menu, and a table of REST endpoints. The table has columns for 'Key', 'GET', 'POST', 'PUT', 'PATCH', 'DELETE', and 'HEAD'. The endpoints listed are: search, stock_availability, stock_movements, stocks, supply_order_details, supply_order_histories, supply_order_receipt_histories, supply_order_states, supply_orders, warehouse_product_locations, and warehouses.

Key	GET	POST	PUT	PATCH	DELETE	HEAD
search	✓					
stock_availability	✓		✓	✓		✓
stock_movements	✓					✓
stocks	✓					✓
supply_order_details	✓					✓
supply_order_histories	✓					✓
supply_order_receipt_histories	✓					✓
supply_order_states	✓					✓
supply_orders	✓					✓
warehouse_product_locations	✓					✓
warehouses	✓	✓	✓	✓	✓	✓

Beispiel: PrestaShop API-Dokumentation

Nachteil beim REST-Zugriff aus FileMaker Sicht ist der meist etwas höhere Programmieraufwand im Vergleich zu ODBC, um die Daten zwischen dem Fremdsystem und FileMaker auszutauschen. Dafür benötigen Sie aber auch keine umständliche ODBC-Konfiguration, sondern nutzen das Standard HTTP(S)-Protokoll, das durch jede Firewall problemlos durchgeht.

Beispiel Ferienhausvermietung

Unsere Ferienhausvermietung-Webseite können wir unverändert aus dem vierten Szenario, also dem Web-Publishing mit SQL, übernehmen. Die Daten belassen wir in der SQL-Datenbank, das Ganze funktioniert ja prächtig.

Zusätzlich erweitern wir die Webseite um eine REST-API und binden FileMaker mittels „Aus URL einsetzen ...“ daran. Wir testen die Kommunikation zwischen FileMaker und der Webseite – alles klappt!

Die FileMaker ODBC-Verbindung wird nun nicht mehr benötigt. Wir löschen den direkten SQL-Datenbankzugang und die Firewall-Regeln, müssen uns um keine SSL-Zertifikate für die ODBC-Verbindung mehr kümmern und können die ODBC-Treiber vom FileMaker Server entfernen. Das System ist schlanker und sicherer geworden, da der direkte Datenbankzugriff auf SQL nicht mehr möglich ist.

Fazit

Es gibt viele Möglichkeiten, wie man FileMaker zum Web-Publishing einsetzen kann. Jede Variante hat ihre Vor- und

Nachteile. Das Schöne ist, dass sich alle Varianten kombinieren lassen und sich nichts ausschließt.

In der Praxis verwende ich fast immer Custom Web Publishing mit SQL. Dazu gibt es, je nach Anforderung, noch eine REST-API. Der Zugriff per FileMaker erfolgt per REST/JSON oder eben per ODBC/ESS. ■

Fußnoten

- 1 <https://github.com/schube/FileMaker-DataAPI-4-Java>
- 2 <https://inter-mediator.org>
- 3 <https://bernhard-schulz.at/filemaker-und-odbc-verbinding-ess-schattentabellen/>
- 4 <https://bernhard-schulz.at>
- 5 <https://github.com/>
- 6 <https://bernhard-schulz.at/?searchsubmit=Search&s=cURL-Beispiele+für+FileMaker+anpassen>

FileMaker Magazin



Das FileMaker Magazin

- Die einzige deutschsprachige Fachzeitschrift zu Claris FileMaker
- Wissen aus erster Hand von anerkannten Claris FileMaker Fachautoren
- Große Themenvielfalt für Anwender und Entwickler

Exklusiv für Premium-Abonnenten

- Sechs FMM Ausgaben pro Jahr
- Kostenlose Nutzung des Abonnentenbereichs auf www.filemaker-magazin.de
- PDF-Archiv mit allen bisher veröffentlichten Ausgaben
- Jede Ausgabe mit kostenlosen Beispieldateien und Zusatzinfos zum Download

Unser Service

- Aktuelle Neuheiten, Tipps und Infos, Kleinanzeigen und vieles mehr jederzeit auf unseren Webseiten
- Hilfe bei allen Fragen zu FileMaker im FMM Forum
- Kompetente Beratung zum Kauf von FileMaker Lizenzen:
Einfach anrufen unter **+49 (0)40 589 65 79 70**.

Eine kostenlose **Leseprobe** des FileMaker Magazins erhalten Sie, wenn Sie hier klicken.

Wenn Sie sich für ein FileMaker Magazin **Abo** interessieren, klicken Sie bitte hier!

Hier finden Sie Aktuelles zu **FileMaker Lizenzen**, egal ob Sie kaufen, mieten oder sich einfach informieren möchten.