Gewusst wie





David Hamann

(Jg. 1986) Geschäftsführer der davidhamann media GmbH. Er entwickelt Software-Lösungen und berät Unternehmen beim Einsatz von FileMaker. Auf Twitter finden Sie ihn unter dem Handle @d_hamann. fmm@davidhamannmedia.com

Nachdruck aus dem FileMaker Magazin siehe auch: www.filemaker-magazin.de

Einmaliger Aufwand für wiederkehrende Routinen

Ein universell einsetzbares Script zur Aktualisierung von Datensätzen

Beim Entwickeln von FileMaker Lösungen finden wir häufig das gleiche Muster für die Aktualisierung von Feldern in Datensätzen: Gehe zu Layout, finde den Datensatz für eine bestimmte ID, setze die Werte in Feld 1, 2 und 3 und gehe zum Ursprungslayout zurück. Während wir das tun, prüfen wir auf Fehler beim Layoutwechsel, stellen sicher, dass der Datensatz gefunden wurde, und dass Felder fehlerfrei gesetzt werden konnten. Da es kaum jemanden gibt, der repetitiv immer die gleichen Dinge tun mag, stelle ich in diesem Beitrag ein kurzes Script vor, mit dem Sie diese Aufgabenstellung auf eine Anpassung von Script-Parametern reduzieren.

Stellen wir uns eine Blackbox vor, die die oben genannte Routine für uns ausführt. Welche Parameter müssten wir mindestens übergeben? Die ID, das Layout bzw. den Kontext, und die zu aktualisierenden Felder inklusive Werte – mehr nicht. Genau diese Parameter sollten wir also auch dem Script übergeben können – und dabei sollte es egal sein, in welchem Kontext wir uns gerade befinden.

Wieso, weshalb, warum?

Mit der Implementierung von "ExecuteSQL ()" (zu deutsch "SQLQueryAusführen") in FileMaker 12 haben viele Entwickler auf die Verfügbarkeit sämtlicher Standardkommandos zur Datenabfrage und -manipulation gehofft: SELECT, UPDATE, DELETE und INSERT INTO. Doch auch in Version 14 unterstützt die Funktion lediglich SELECT-Statements¹ und hilft uns daher nicht bei der Aktualisierung von Datensätzen. Hauptgrund dafür ist, dass FileMaker Kalkulationen in einer nahezu "reinen funktionalen Sprache"² geschrieben werden (Ausnahme "Let" bzw. zu deutsch "SetzeVar"). Ob wir in Zukunft andere Möglichkeiten der Datenmanipulation erhalten werden, bleibt offen. Da wir es uns aber schon heute etwas einfacher machen möchten, kapseln wir Layoutwechsel, Suche und Aktualisierung in einer Routine und nutzen ab diesem Zeitpunkt nur noch den Aufruf dieser Routine.

Ein wichtiger Hinweis, bevor wir in die Erstellung des Scripts einsteigen: Die beschriebene Methode funktioniert nur, wenn Sie strikten Namenskonventionen für Ihre ID-Felder und Layouts folgen – Tipps zu Benennungsstandards finden Sie in vielen früheren Ausgaben des FileMaker Magazins.

Erstellung der "Black Box"

Das zu erstellende Script, nennen wir "core.pub.api.setField-ForPK (pk ; layoutCode ; fieldList ; fieldValueN)"– benennen Sie es gerne nach Ihrem präferierten Standard um.

Wir werden zwei weitere generische Scripts sowie Eigene Funktionen für das Parameter-Handling einsetzen. Diese können Sie einfach aus der Beispieldatei kopieren und ggf. auf Ihre Namenskonventionen anpassen.

Wie zu Beginn beschrieben, benötigen wir vier Parameter, die wir wie folgt benennen:

- **pk** ist der primäre Schlüssel (ID) des Datensatzes, den Sie aktualisieren möchten.
- layoutCode ist die Schlüsselbezeichnung des Layouts, auf dem die Aktion ausgeführt werden soll. Hier empfiehlt es sich, ein eigenes Layout zu erstellen, das lediglich für die Script-Ausführung genutzt wird und keine Felder/kein Design enthält.



fieldList ist eine Liste der Felder, deren Werte verändert werden sollen. Stellen Sie sicher, dass Sie die Feldnamen in der Liste jeweils mit der "GetFieldName()"-Funktion (zu deutsch " HoleFeldname()") erstellen. Damit verhindern Sie mögliche Fehler bei einer späteren Umbenennung eines Feldes. (Auch wenn FileMaker die Umbenennung von Feldern weitestgehend ohne ernsthafte Konsequenzen erlaubt, ist es eine gute Angewohnheit, möglichst darauf zu verzichten.)

Nachdruck aus dem

FileMaker Magazin siehe auch: www.filemaker-magazin.de

fieldValueN ist der Hinweis auf einen Parameter, der mehrfach auftreten kann (fieldValue1, fieldValue2, ...), da für jedes Feld in fieldList ein Wert übergeben werden muss. Eine getrennte Übergabe (d. h. nicht als Liste), hat den Vorteil, dass a) Werte selbst Listen sein können und b) ein expliziter Datentyp gesetzt werden kann.

1	Fehleraufzeichnung setzen Option: Ein
2	AnwenderAbbruchZulassen setzen Option: Aus
3	
4	Variable setzen Name: \$result Wert: Falsch
5	
6	# Hole Parameter
7	Variable setzen Name: \$pk Wert:getParam(_sp; "pk")
8	Variable setzen Name: \$layoutCode Wert:getParam (_sp ; "layoutCode")
9	Variable setzen Name: \$fieldList Wert:getParam(_sp;"fieldList")

Herstellen des Kontexts

Nun stehen die übergebenen Parameter zur Verfügung (auf fieldValueN kommen wir später noch zurück) und wir können zum Layoutwechsel übergehen. Dazu rufen wir ein Script auf, das nach einem Layout sucht, welches den übergebenen layoutCode in eckigen Klammern im Namen enthält.

In der Beispieldatei nutzen wir zur Vereinfachung nur ein Layout. Wie oben beschrieben macht es in einem Produktionssystem Sinn, ein dediziertes Layout für die Scripts zu verwenden, damit keine Benutzeroberfläche geladen werden muss.

Wir prüfen, ob der Layoutwechsel erfolgreich war und suchen dann den korrekten Datensatz, der der übergebenen ID entspricht (Parameter **pk**). Hierzu nutzen wir ein universal einsetzbares Script "core.pub.nav.findPK()" (siehe Beispieldatei). Wenn wir bereits auf dem richtigen Datensatz stehen, sparen wir uns die Suche.

Sollten Sie andere Namenskonventionen für ID-Felder verwenden, passen Sie bitte die Eigene Funktion "_pkName()" an.

▼
▼
٠

Aktualisierung des Datensatzes

Jetzt kommen wir zum spannenden Teil: Für jedes Feld in **\$fieldList** setzen wir das entsprechende Feld mit dem Wert von **fieldValueN** (N bezieht sich auf den Listenwert von **\$fieldList**, d.h. **fieldValue1** ist der Wert für den ersten Eintrag in **\$fieldList**, **fieldValue2** für den zweiten Eintrag, usw.).

Kommt es beim Setzen eines Feldes zu einem Fehler, erhöhen wir den **\$errorCount**. Nur wenn **\$errorCount** am Ende gleich 0 ist, geben wir ein Ergebnis von "True" (zu deutsch "Wahr") zurück.

1	Schleife (Anfang) 🗸 🔻
2	Verlasse Schleife wenn ◆ Wert: SetzeVar(\$i=\$i+1;\$i>\$iMax)
3	
4	Variable setzen Name: \$fieldName Wert: HoleWert (\$fieldList;\$i)
5	
6	Feld nach Namen einstellen Name: \$fieldName Wert: getParam(_sp;" fieldValue" & \$i)
7	Variable setzen Name: \$errorCount Wert: Falls(Hole(LetzteFeblerNr)=0; \$errorCount; \$errorCount+1)
8	Schleife (Ende)

Und das ist alles! Der Datensatz ist (hoffentlich) aktualisiert und das "Elternscript" erhält eine ordnungsgemäße Rückmeldung.

Schlusswort

Ein Wort zum Thema Kontext: In der Regel laufen diese modularen Scripts über "Perform Script On Server" (zu deutsch "Script auf Server ausführen"), sodass eine Veränderung der Fundmenge keine wirkliche Rolle spielt. Möchten Sie das Script direkt auf dem Client laufen lassen, greifen Sie auf Techniken wie Off-screen-Fenster zurück oder speichern Sie die Fundmenge in einem zweiten Tabellenauftreten (Wiederherstellung mit "Go To Related Records" / "Gehe zu Bezugsdatensatz").

Ich hoffe, diese wenigen Zeilen helfen Ihnen dabei, mehr Zeit in die kniffligen Herausforderungen Ihrer Lösung zu stecken und weniger Zeit mit Fleißarbeit zu verbringen.

Fußnote

- ¹ Mit Plugins kann diese Limitierung umgangen werden.
- ² https://en.wikipedia.org/wiki/Purely_functional



Magazin

Das FileMaker Magazin

- Einzige, deutschsprachige Fachzeitschrift zu FileMaker
- Wissen aus erster Hand von anerkannten FileMaker Fachautoren
- Große Themenvielfalt f
 ür Anwender, Entwickler und Fortgeschrittene

Exklusiv für Premium-Abonnenten

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

Unser Service

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

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

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!

K&K Verlag GmbH · www.filemaker-magazin.de · Telefon +49 40 589657970