Erschienen im
FileMaker Magazin
www.filemaker-magazin.de

Ronny Schrader
(Jg. 1971), Examen im Gesundheitswesen, Studium der Wirtschaftsinformatik, Vertrieb von Pflege-Software, eigene Pflegedienste, seit FileMaker 7 Entwickler im Gesundheitsbereich, seit 2019 MaRo-Programmierung GbR, Fokus auf Performance, Integration und Benutzerfreundlichkeit.
ronny@filemaker-experts.de

Link **Webserver mit PHP**

Sequentielle Suche

Arbeitsteilung zwischen FileMaker und PHP

FileMaker ist bekannt für seine exzellente Suchfunktion. Suchen können zusammengefasst, erweitert und tabellenübergreifend durchgeführt werden. Allerdings fehlt die Möglichkeit, die Suche zeichenweise zu verfeinern, ähnlich wie im Browser, wo die Ergebnisse mit jedem eingegebenen Zeichen präziser werden.

In FileMaker lassen sich zwar diverse Ansätze umsetzen, um eine sequentielle Suche zu realisieren, etwa mit globalen Feldern, der Schnellsuche oder mit Triggern. Allerdings wird diese Methode bei größeren Datenmengen zunehmend langsamer. Bei einem Kunden funktionierte dieser Ansatz jahrelang einwandfrei – bis plötzlich bei jedem Tastendruck der Cursor im Suchfeld neu positioniert werden musste. Dieser Zustand war unzumutbar und sollte innerhalb weniger Tage behoben werden. Die Lösung für das Problem kam mir, wie so oft, bei einem nächtlichen Spaziergang mit meinem Hund.

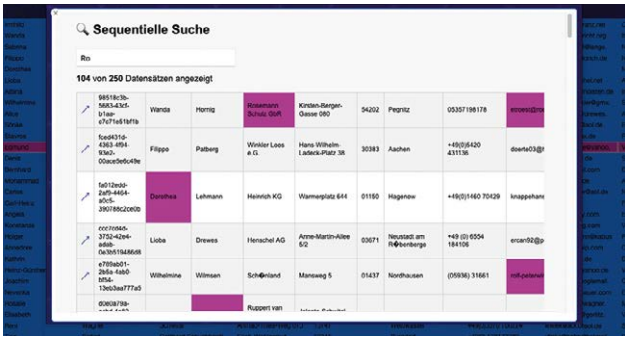
Wenn „Motorsteuerung“ nicht gleich „Motor-Steuerung“ ist

Bei der sequenziellen Suche wird Zeichen für Zeichen geprüft. Sie ist weder feldbasiert noch indexgestützt, sondern arbeitet direkt mit der gesamten Zeichenkette. Bei dieser Methode wird eine Suchanfrage linear mit jedem Datensatz verglichen, ähnlich wie bei einem reinen Textvergleich. Dabei ist nur relevant, ob die gesuchte Zeichenkette irgendwo im Text vorkommt. Genau dieses Verhalten benötigen wir. Der Kunde weiß beispielsweise, dass er einen Notizeintrag mit dem Begriff „Motorsteuerung“ hinterlegt hat. Er ist sich jedoch nicht sicher, ob er eventuell „Motor-Steuerung“ geschrieben hat. Diese Frage lässt sich in FileMaker nur schwer beantworten. Über ein PHP-Script ist eine Lösung möglich, die performant ist und sich ansprechend gestalten lässt.

Der WebViewer als Suchfenster

Die Idee ist folgende: Beim Betreten des Suchfensters werden die Daten einer Tabelle per POST an ein PHP-Script übermittelt. Das lässt sich mit FileMaker relativ einfach umsetzen. Die Daten werden als JSON-Objekt im Body eines HTTP-POST-Requests an das Script übergeben.

Allerdings habe ich die Menge der zu übertragenden Felder anfangs unterschätzt und mich schließlich für einen pragmatischeren und einfacheren Weg entschieden. Ich exportiere die Tabellendaten als Datei und lade sie gesammelt per POST-Aufruf auf den Server.



Dateiübertragung aus FileMaker

Die gesamte Tabelle wird zunächst in eine temporäre CSV-Datei exportiert und anschließend per cURL an das PHP-Script auf dem Server übergeben. Der Pfad wird dynamisch erzeugt und in eine Variable geschrieben, die anschließend als Upload-File an den cURL-Befehl übergeben wird. Das FileMaker Script hat folgenden Aufbau:

```
◆ # Temporäre Datei erzeugen
◆ Variable setzen
  [ $exportPath ; Wert: "file:" & Hole ( TemporärerPfad ) &
  "projekt.csv" ]
◆ Datensätze exportieren
  [ Mit Dialog: Aus ; Ordner erstellen: Ein ; "$exportPath" ;
  Macintosh ]
◆ # Pfad für cURL aufbereiten (Mac-Spezifikum *)
◆ Variable setzen
  [ $filePathForCurl ; Wert: Austauschen ( $exportPath ;
  "Macintosh HD" ; "private" ) ]
◆ # Upload vorbereiten
◆ Variable setzen
  [ $$curlOptions ; Wert:
    "--request POST" &
    " --upload-file $file" &
    " --header \"Content-Type: text/csv\"" ]
◆ Variable setzen
  [ $file ; Wert:demodaten-adressen::csv_upload ]
◆ # Kurze Scriptpause
◆ Scriptpause setzen
  [ Dauer (Sekunden): ,2 ]
◆ # URL Festlegen
◆ Variable setzen
  [ $url ; Wert: "https://maps.maro-test.de/FM/suche.php" ]
◆ # Datei übertragen
◆ Aus URL einfügen
  [ Ziel: $$response ; $url ; cURL-Optionen: $$curlOptions ]
◆ # Die Anzeige der Daten im WebViewer vorbereiten
◆ Variable setzen
  [ $$URL_VIEWER ; Wert: $$response ]
◆ # Ein Suchfenster öffnen
◆ Neues Fenster
  [ Stil: Karte ; Name: "add_action" ; Mit Layout: "Suche"
  (demodaten-adressen) ]
◆ Objekt aktualisieren
  [ Objektname: "URL_VIEWER" ; Wiederholung: 1 ]
```

Wichtiger Hinweis:
Der Exportpfad `(Hole(TemporärerPfad))` erzeugt einen gültigen Pfad im jeweiligen Betriebssystem – auf macOS z. B. ***file:/Macintosh HD/private/var/...***, unter Windows z. B. ***file:/C:/Users/...***.

CSV-Datei empfangen

Das PHP-Script ist so gestaltet, dass es Tabellen mit einer beliebigen Anzahl von Feldern empfangen kann, egal ob 20 oder 100. Auch die Feldbezeichner sind ohne Relevanz. Für die Verarbeitung der wiedergegebenen Werte ist es nur wichtig, dass die erste Spalte den Suchschlüssel für FileMaker enthält.

Bei Adressen ist das die **ID_Adresse** und bei Aufträgen die **ID_Auftrag**.

Die vom FileMaker System per POST hochgeladene CSV-Datei wird auf serverseitig direkt über ***php://input*** eingelesen, in Zeilen aufgeteilt und als Array verarbeitet. Dabei spielt es keine Rolle, ob die Datei unter Windows oder macOS erzeugt wurde. Dank `preg_split()` ist das Script betriebssystemunabhängig. Das Script hat folgenden Aufbau:

```php
<?php
$csvData = [];
// Dateiinhalt direkt einlesen
$data = file_get_contents("php://input");

if ($data && strlen($data) > 0) {
   // Robust gegen \r\n (Windows), \n (Unix), \r (Mac)
   $lines = preg_split('/\r\n|\r|\n/', $data);

   foreach ($lines as $line) {
      if (trim($line) !== "") {
         $csvData[] = str_getcsv($line, "\t"); //
         Tab-getrennt
      }
   }
   if (!empty($csvData) && empty(array_
      filter(end($csvData)))) {
      array_pop($csvData);
   }
}
// Alle Spalten dynamisch ermitteln
$spaltenIndizes = range(0, count($csvData[0] ?? []) -
1);
?>
```
PHP

Rückgabe mehrerer Werte an FileMaker

Sobald der Benutzer im Suchfeld die Eingabetaste drückt, werden alle aktuell sichtbaren Einträge, die durch den Filter gefunden wurden, erfasst. Aus der ersten sichtbaren Spalte wird die eindeutige ID extrahiert. Diese IDs werden in einer Liste zusammengefügt, die mit dem Operator „|“ (Pipe) getrennt und per ***fmp://-URL*** an ein FileMaker Script übergeben wird. Dies könnte auch anders gelöst werden, allerdings hat sich das Pipe-Zeichen in den letzten 30 Jahren als sehr robust erwiesen.

```
document.getElementById("searchInput").
addEventListener("keypress", function(event) {
   if (event.key === "Enter") {
      event.preventDefault();
      const visibleRows = document.
querySelectorAll("#csvTable tbody tr:not(.hide)");
      const ids = [];

      visibleRows.forEach(row => {
         const cells = row.querySelectorAll("td");
         const id = cells[1]?.textContent.trim(); //
ID in Spalte 1
         if (id) ids.push(id);
      });

      if (ids.length > 0) {

         const param = encodeURIComponent(ids.
```

```
join("|"));
          window.location.href = `fmp://$/
Suche?script=Suche_FileMaker&param=${param}`;
      } else {
          alert("Keine Einträge gefunden.");
      }
   }
});
```
PHP

Direkter Aufruf eines Eintrages

Natürlich kann aus einer Ergebnismenge auch ein einzelner Datensatz gewählt werden. Dafür enthält jede Zeile einen Link, über den sich der Datensatz aufrufen lässt.

```
<a href="fmp://$/Suche?script=Suche_
FileMaker&param=<?= urlencode($row[0] ?? '') ?>">↗</
a>
```
PHP

Das gesamte Script liegt der Demo-Datei bei. Dort ist auch der HTML-Aufbau ersichtlich.

Wie geht es in FileMaker weiter?

Wie aus dem PHP-Script ersichtlich ist, gibt es zwei Möglichkeiten, die Werte an FileMaker zurückzugeben. Das Ergebnis kann entweder als Menge mit mehreren Datensätzen oder gezielt als ein Datensatz vorliegen. Zur Realisierung der Suche wurde in meinem Beispiel ein sehr einfacher Ansatz innerhalb von FileMaker verwendet.

Die Übergabe erfolgt als einfacher String, der alle gefundenen IDs enthält, getrennt durch das Pipe-Zeichen (|).

Dieses Format ist leicht zu verarbeiten und eignet sich gut für eine schnelle Mehrfachsuche in FileMaker. Bei Bedarf besteht die Möglichkeit, die Trennung an die individuellen Anforderungen anzupassen. Sollte das Pipe-Zeichen innerhalb von Textfeldern vorkommen, empfiehlt sich ein Wechsel auf ein anderes Trennzeichen oder die Übergabe per JSON. Dadurch können potenzielle Konflikte zuverlässig vermieden werden.

```
◆ # Hier holen wir die ID's aus dem Parameter
◆ Variable setzen
  [ $liste ; Wert: Hole ( ScriptParameter ) ]
◆ Variable setzen
  [ $elemente ; Wert: Austauschen ( $liste ; "|" ; "¶" ) ]
◆ # Schließe das Suchfenster
◆ Fenster schließen
  [ Name: "add_action" ; Aktuelle Datei ]
◆ #
◆ # Hier eine einfache Suche, die kann jeder nach seinen
    Vorstellungen umsetzen
◆ # Anzahl der gefundenen Datensätze ermitteln
◆ Variable setzen
  [ $anzahl ; Wert: ElementeAnzahl ( $elemente ) ]
◆ Variable setzen
  [ $loop ; Wert: 1 ]
◆ Variable setzen
  [ $id ; Wert: ElementeMitte ( $elemente ; $loop ; 1 ) ]
◆ # An dieser Stelle ist bei Parametersteuerung auch eine
    andere Tabelle möglich
◆ Ergebnismenge suchen
  [ Wiederherstellen ]
◆ Wenn
  [ $anzahl > 1 ]
◆ Schleife (Anfang)
  [ Flush: Immer ]
◆ Variable setzen
  [ $loop ; Wert: $loop  + 1 ]
◆ Variable setzen
  [ $id ; Wert: ElementeMitte ( $elemente ; $loop ; 1 ) ]
◆ Ergebnismenge erweitern
  [ Wiederherstellen ]
◆ Verlasse Schleife wenn
  [ $loop ≥ $anzahl ]
◆ Schleife (Ende)
◆ Ende (wenn)
```

Systemeinrichtung

Für die Realisierung der Lösung wird ein Webserver mit einer Standard-PHP-Umgebung benötigt. Als Beispiele seien hier **Apache** oder **Nginx** mit PHP ab Version 7.0 empfohlen. Bitte beachten Sie, dass die PHP-Funktionen „file_get_contents()“ und „curl“ aktiviert sein müssen. Eine Datenbank ist nicht erforderlich, da die Suche vollständig dateibasiert erfolgt.

Der Webserver muss per HTTPS erreichbar sein, da FileMaker (insbesondere ab Version 19) bei der Kommunikation zunehmend auf gesicherte Verbindungen besteht.

Auf der FileMaker Seite genügt bereits eine Einzelplatzlösung. Das System lässt sich ebenso problemlos in einer Server-Umgebung einsetzen. Wichtig ist die Aktivierung von „URLs dürfen FileMaker-Scripts ausführen (fmuriscript)“ in den erweiterten Zugriffsrechten.

Im FileMaker Script zur Datenübertragung wird das globale Feld **csv_upload** verwendet. Es muss nicht zwingend Bestandteil der zu exportierenden Tabelle sein, sondern lediglich für das Script erreichbar sein. Das kann beispielsweise über eine Kontextbeziehung oder ein zentrales Layout erfolgen. So bleibt das Setup flexibel und lässt sich mit minimalem Aufwand in bestehende FileMaker Scripts integrieren.

Fazit

Die Ergebnismenge erhalten wir wie gewünscht direkt innerhalb von FileMaker – schnell, zuverlässig und ohne strukturelle Einschränkungen. Die Suche ist dabei äußerst flexibel, da wir uns kaum Gedanken über die konkrete Tabellenstruktur machen müssen.

Besonders wichtig ist es, die zu exportierenden Felder im Übertragungsscript gezielt festzulegen. Wenn die Suche über Parameter gesteuert wird, können wir an dieser Stelle dynamisch bestimmen, welche Felder aus welcher Tabelle exportiert werden sollen.

Das Script, das aus der PHP-Datei heraus in FileMaker aufgerufen wird, bleibt dabei bewusst schlank und lässt sich bei Bedarf ebenso dynamisch erweitern wie der Export selbst.

Bei großen Datenmengen empfiehlt es sich, den Export als Serverscript auszuführen. Dadurch bleibt auch bei mehreren tausend Datensätzen die Performance erhalten. ■