

## Analyse des cURL-Befehls

Sehen wir uns den Login-Vorgang per cURL einmal im Detail an:

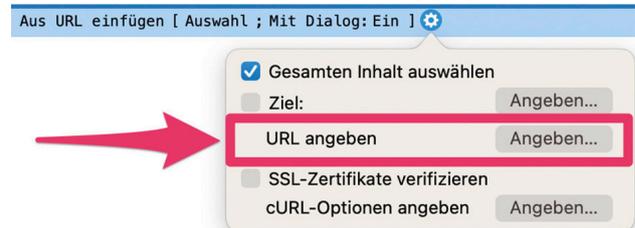
```
## Login
curl -X POST \
  https://hobexplus.brunn.hobex.at/api/account/login \
  -H 'Content-Type: application/json' \
  -d '{
    "userName": "schulz",
    "password": "XXXXXX"
  }'
```

cURL

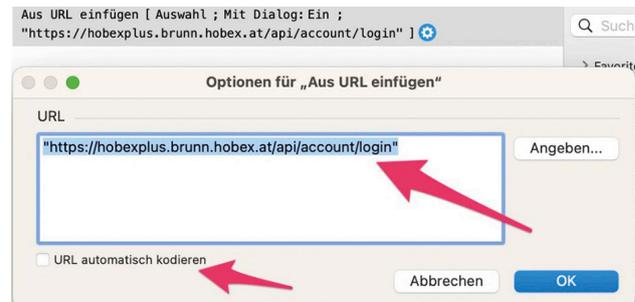
- **curl**  
Der Name des Programms, damit das Betriebssystem (im macOS **Terminal** oder Windows **Eingabeaufforderung**) weiß, welches Programm ausgeführt werden soll.
- **-X POST**  
Wir senden eine HTTP-POST-Nachricht. Lässt man das **-X POST** weg, wird automatisch eine HTTP-GET-Nachricht geschickt. Ob an dieser Stelle ein POST, ein GET oder ein DELETE, PUT, PATCH, HEAD etc. benötigt wird, ist von der verwendeten API abhängig und dort dokumentiert. Im Falle des **Hobex**-Logins ist ein POST notwendig.
- **https://hobexplus.brunn.hobex.at/api/account/login**  
Die URL, an die der Befehl gesendet wird.
- **-H 'Content-Type: application/json'**  
Mit unserer Anfrage senden wir den HTTP-Header Content-Type mit dem Wert **application/json**. Statt **-H** kann man auch die längere Schreibweise **--header** verwenden. Welche Header gesendet werden müssen, ist wieder der jeweiligen API-Dokumentation zu entnehmen.
- **-d '{"userName": "schulz","password": "XXXXXX"}'**  
Hier werden die Daten an den Server gesendet – in unserem Fall ein JSON-Dokument. An dieser Stelle können aber auch Dateien hochgeladen, XML-Dokumente oder Formular-Daten gesendet werden etc. Auch hier gilt: Was der Server erwartet, beschreibt die API-Dokumentation der Schnittstelle.

## Umsetzung in FileMaker

Nachdem wir nun wissen, wie der cURL-Befehl aufgebaut ist, soll das Ganze zu FileMaker in den Befehl: „Aus URL einfügen“ übertragen werden.

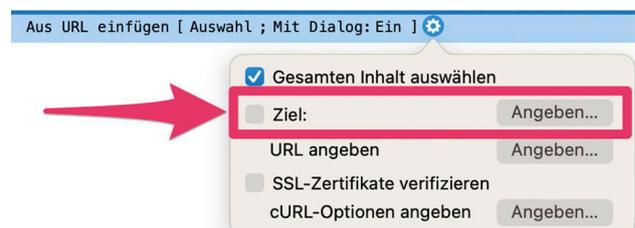


Als Erstes definieren wir die URL, die bei uns ein fester Text ist und daher in Anführungszeichen in den entsprechenden Dialog eingetragen wird. Theoretisch könnte man sie auch per FileMaker Formel berechnen, aber das ist bei unserem einfachen Beispiel nicht notwendig.



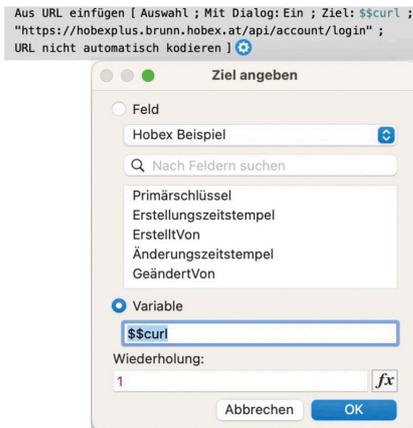
Im Dialog gibt es die Option, die URL automatisch zu kodieren. Ist das Kennzeichen aktiviert, wird eine URL wie `https://www.schubec.com/?Das` ist ein Beispiel, verständlich automatisch in `https://www.schubec.com/?Das%20ist%20ein%20Beispiel%2C%20verst%C3%A4ndlich` umgewandelt: Leerzeichen, Umlaute und andere Sonderzeichen werden dann korrekt URL-kodiert.

Da ich es nicht mag, wenn das automatisch passiert, deaktiviere ich diese Zusatzoption immer. Wenn ich Bedarf an der Funktionalität habe, würde ich das explizit mit dem FileMaker Befehl „LiesAlsURLVerschlüsselt“ lösen.



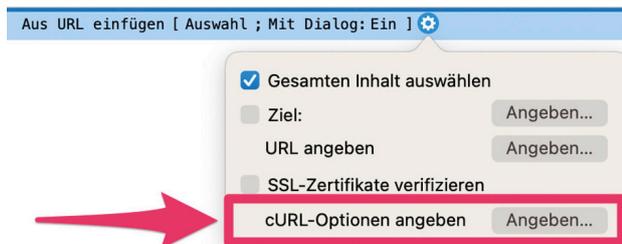
Als Nächstes wird das Ziel festgelegt, also das Feld oder die Variable, in der wir die Antwort des Servers in FileMaker weiterverarbeiten möchten. Ich speichere das Ergebnis in der globalen Variablen `$$curl`. Obwohl ich üblicherweise kein großer Freund von globalen Variablen bin, hat das in diesem Fall den Vorteil, dass man in Scripts mit dem cURL-Ergebnis gut weiterarbeiten kann und dass man im FileMaker Data Viewer

schnell den Inhalt einsehen kann.



## cURL-Optionen

Kommen wir nun zu den cURL-Optionen und damit zum herausforderndsten Teil.



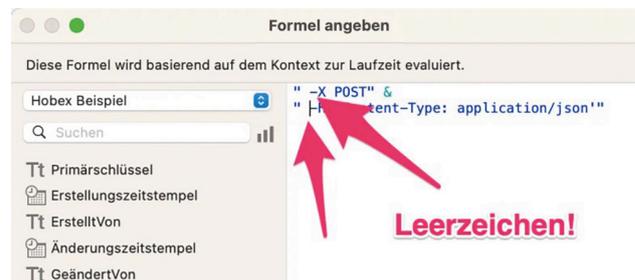
Den ersten Teil unseres cURL-Terminal-Befehls – das Senden der **POST**-Anfrage – können wir einfach in FileMaker übernehmen. Die Option ist als Text in FileMaker einzutragen, daher müssen am Anfang und am Ende des Befehls Anführungszeichen stehen:



Auch die nachfolgende Header-Angabe „-H 'Content-Type: application/json'“ können wir als Text (also mit Anführungszeichen) im FileMaker Dialog ergänzen. Die beiden Texte werden wie gewohnt mit dem „&“-Zeichen verbunden:



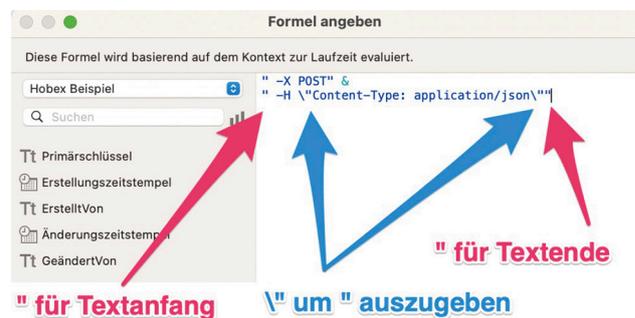
Leider funktioniert das Ganze aber nicht so wie gewünscht. Das erste Problem wird vermutlich schnell sichtbar: Wenn man die beiden Texte verknüpft, kommt als Ergebnis `-X POST-H 'Content-Type: application/json'` heraus: **POST** und **-H** stehen ohne Zwischenraum nebeneinander. Das lässt sich natürlich sehr einfach lösen, indem man nach dem **POST** ein Leerzeichen ergänzt. Allerdings ist das bei umfangreichen Formeln mit mehr als zehn Zeilen eher unübersichtlich, weshalb ich mir angewöhnt habe, immer am Anfang der Zeile ein Leerzeichen einzufügen. Das mache ich auch bei dem ersten Befehl, denn so kann ich später die Reihenfolge der Zeilen problemlos ändern, ohne an Leerzeichen denken zu müssen:



Das zweite Problem ist ziemlich verzwickelt und es hat mich viele Stunden gekostet, bis ich die Ursache erkannt habe.

Fast alle cURL-Beispiele im Internet haben die Header-Angabe in der Form `-H 'Content-Type: application/json'`. Unsere „Diva“ FileMaker akzeptiert das aber nicht und erwartet anstelle der einfachen Hochkommata (') die doppelten Hochkommata ("). Um diese Angabe in FileMaker zu übersetzen, muss man sie also folgendermaßen schreiben: `-H "Content-Type: application/json"`. Wenn man das weiß, ist es kein Problem, aber wissen muss man es halt!

Ja, und weil FileMaker auch Texte in Hochkommata einschließt und deshalb nicht weiß, wo der Text anfängt und aufhört, müssen wir die Hochkommata der Header-Angabe mit dem Backslash escapen<sup>3</sup>. Somit wird dann Folgendes daraus:



Fehlen noch die Daten selbst, also der Teil `-d '{"userName": "schulz", "password": "XXXXXX"}'`. Hier senden wir ein JSON-Dokument an den Server. JSON-Dokumente können äußerst komplex sein und viele Sonderzeichen enthalten, sodass es extrem mühsam wäre, diese immer per Hand „zusammenzubauen“. Glücklicherweise bietet FileMaker eine einfache Funktion zum Versenden von JSON-Dokumenten:

Zuerst stellen wir in einer Variable namens `$json` unser JSON-Dokument mit dem FileMaker Befehl „JSONSetElement“ zusammen.

```
//Wir fangen mit einem leeren JSON-Dokument an:
JSONSetElement ( "" ;
//Wir ergänzen userName als String/Text:
["userName"; "Schulz"; JSONString];
//Wir ergänzen password als String/Text:
["password"; "xxxxxxx"; JSONString ] )
```



Dieses JSON-Dokument können wir nun wie folgt beim cURL-Befehl ergänzen:



Das hat den großen Vorteil, dass sich FileMaker selbst um das Escapen von Sonderzeichen im JSON-Dokument kümmert und wir die Daten einfach per Referenz auf die Variable (`@$variablenname`) senden können. Dokumentiert ist diese Schreibweise inkl. aller Optionen hier: <https://help.claris.com/en/pro-help/content/curl-options.html> – die Variable kann einen beliebigen Namen tragen, muss also nicht zwingend `$json` heißen.

## Fehlerhandling

Nun kann es passieren, dass etwas nicht klappt – vielleicht ist der Server nicht erreichbar, der Content-Type wurde falsch gesetzt, der Parameter fürs Passwort ist nicht korrekt oder Ähnliches. Es ist sinnvoll, für diese Fälle vorzusorgen, deshalb ergänzen wir bei den cURL-Optionen das FileMaker Fehlerhandling `--show-error` wie folgt:



Interessante Befehle zum Fehlersuchen sind auch `--dump-header` und `--trace-ascii` – dafür werde ich aber einmal einen gesonderten Artikel schreiben.

`--show-error` sorgt dafür, dass der FileMaker Befehl „Hole ( LetzteFehlerNummer )“ im cURL-Fehlerfall den Wert „1631“ liefert, sodass wir darauf reagieren können. Warum „1631“? Weil das unter <https://help.claris.com/en/pro-help/content/curl-options.html> genau so festgelegt ist. Weitere FileMaker Fehlernummern findet man unter <https://help.claris.com/en/pro-help/content/error-codes.html>

Um genau herauszufinden, was schief gelaufen ist, nutzen wir die Funktion „Hole ( LetzteFehlerNrDetail )“, die uns Details zum Fehler liefert. Unter <https://curl.se/libcurl/c/libcurl-errors.html> können Sie nachschauen, welche cURL-Fehler es gibt.

cURL gibt oft die HTTP-Fehlercodes zurück, wie z. B. die „404“, wie Sie in der Abbildung weiter unten sehen können. Das sind HTTP-Statuscodes, deren jeweilige Bedeutung unter <https://de.wikipedia.org/wiki/HTTP-Statuscode> nachzulesen ist.

In der Praxis finden wir mit dem FileMaker „1631“-Fehlercode heraus, ob unsere cURL-Abfrage geklappt hat oder eben nicht.

Ist der Fehlerdetailcode kleiner als 100, handelt es sich ziemlich sicher um einen cURL-Fehler, liegt er darüber, spricht viel für einen HTTP-Statuscode. Mit diesem Wissen ausgestattet und ein wenig Übung lassen sich die Fehler rasch beheben. Wir ergänzen unser Script daher wie folgt:

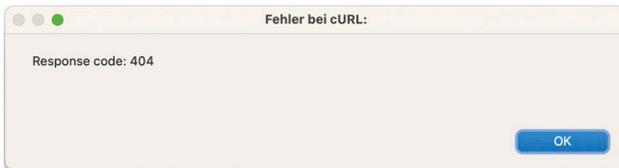


Wenn Sie das Script ausführen (und vorher noch die Option „Mit Dialog: aus“ bei „Aus URL einfügen aktivieren“), erhalten Sie:



Technisch hat die Anfrage geklappt und es kam eine korrekte Antwort vom Server retour. Dass inhaltlich ein Fehler vorliegt (Passwort falsch), ist ein anderes Thema, das in jeder API ein wenig anders behandelt wird. Hätte ich diese *Hobex*-API entworfen, hätte ich bei einem falschen Passwort einen „HTTP-Fehlercode 401/Unauthorized“ zurückgeliefert. Aber das handhaben die API-Hersteller eben unterschiedlich und das macht die Sache manchmal auch etwas komplizierter.

Wenn wir die Anfrage einmal absichtlich falsch absenden, beispielsweise an die nicht existierende URL <https://hobexplus.brunn.hobex.at/api/account/login2>, erhalten wir eine entsprechende Fehlermeldung:



So sieht übrigens die Antwort des Servers aus, wenn man das korrekte Passwort angibt:



Mit `JSONGetElement ( $$curl ; "token" )` kann man den Token auslesen und für weitere Aufrufe verwenden. Hier im *Hobex*-Beispiel, um eine Kreditkartenzahlung mit einem zweiten CURL zu veranlassen. Sie finden es abgedruckt ein paar Seiten zurück auf Seite 16 unten rechts.

## Beispieldatei zum Download

Damit Sie nicht alles abtippen müssen, können Sie hier die fertige Datei herunterladen: <https://bernhard-schulz.at/wp-content/uploads/2023/06/Hobex-Login-per-FileMaker-cURL.fmp12.zip>

## Wie geht es weiter?

Ich persönlich finde es sehr wichtig, dass man mit den Grundlagen vertraut ist, cURL-Befehle aus dem Internet in FileMaker übersetzen kann und dabei auch halbwegs versteht, was man macht und nicht nur blind irgendetwas abtippt.

Sobald man die Grundlagen verstanden hat, darf man sich das Leben aber leichter machen und Tools verwenden, die beim „Übersetzen“ helfen. Zum Beispiel:

- <https://www.soliantconsulting.com/blog/translating-auto-generated-curl-to-filemaker-curl/>
- <https://dbservices.com/blog/filemaker-curl-options>

Selbstverständlich finden Sie online noch viele weitere Beispiele, die allerdings meist englischsprachig sind. Hier eine kleine Auswahl:

- <https://mossrock.com/intro-to-curl-filemaker/>
- <https://skeletonkey.com/an-almost-universal-api-template-for-claris-filemaker-and-filemaker-pro/>

## Kompatibilität

Das Schöne ist, dass der Befehl „Aus URL einfügen“ auf der gesamten *Claris*-Plattform zur Verfügung steht, sodass Sie

das Erlernte mit FileMaker Pro, FileMaker Go, FileMaker Server und FileMaker Web Direct nutzen können.

## Feedback

Hat Ihnen dieser Artikel gefallen? Ich freue mich über Ihre Rückmeldung per E-Mail

## Fußnoten

- 1 FMM\_202302, S. 19ff
- 2 <https://everything.curl.dev/project/usershttps://everything.curl.dev/project/users>
- 3 <https://de.wikipedia.org/wiki/Escape-Sequenz>

# 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](http://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 Websites
- 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.