Using Apple's Global Service Exchange web service in FileMaker

Quite a few Apple shops use FileMaker and/or Xojo for their development of in-house tools. A common request is to use Apple's webservices to query warranty status. So today I want to show some scripts on how to do this with the newer REST API. First of course you have to ask Apple for a GSX login which may require some paperwork. Next you need to white list your static IP for their webservice and get the credentials.

You request a certificate from Apple, so you generate a private key. The tricky key is to copy the private key with the certificate into one pem file. This pem file is than used with our script. Also please download a standard cacert.pem file with root certificates.

Here is an example script to run a query, e.g. to get the authentication token:

```
# What to do, maybe passed as parameter
Set Variable [ $Query ; Value: "authenticate/token" ]
Set Variable [ $JSON ; Value: "" ]
#
# Some constants you may have
Set Variable [ $shipToCode ; Value: "XXXXXXXXXX" ]
Set Variable [ $soldToCode ; Value: "XXXXXXXXXX" ]
Set Variable [ $userIDCode ; Value: "xxxxxxxx@xxxxxxxx.com" ]
Set Variable [ $ServerURL ; Value: "https://partner-connect-uat.apple.com/gsx/
api/" ]
Set Variable [ $PrivateKeyPassword ; Value: "xxx" ]
#
# Where are key files stored?
If [ MBS("IsServer") ]
    # Folder on server with key files
    Set Variable [ $path ; Value: "/Library/FileMaker Server/Data/GSX/" ]
Else
    # For local testing also on developer's computer
    Set Variable [ $path ; Value: "/Users/admin/Documents/GSX/" ]
End If
```

```
#
# Query
Set Variable [ $curl ; Value: MBS("CURL.New") ]
#
# -------------------------------------------------------------------------------------
# Build final URL from Server and whatever query you need
Set Variable [ $r ; Value: MBS("CURL.SetOptionURL";$curl; $ServerURL &
$Query) ]
#
# We pass PEM file with private key and certificate together
Set Variable [ $r ; Value: MBS("CURL.SetOptionSSLCertType"; $curl; "PEM") ]
Set Variable [ $r ; Value: MBS("CURL.SetOptionKeyPassword"; $curl;
$PrivateKeyPassword) ]
Set Variable [ $r ; Value: MBS("CURL.SetOptionSSLCert"; $curl; $pfad &
"cert.pem") ]
# The usual CURL cacert.pem with valid root certificates
Set Variable [ $r ; Value: MBS("CURL.SetOptionCAINFO"; $curl; $pfad &
"cacert.pem") ]
# We want TLS 1.2
Set Variable [ $r ; Value: MBS("CURL.SetOptionSSLVersion"; $curl; 6) ]
# Wait maximum 10 seconds for answers
Set Variable [ $r ; Value: MBS("CURL.SetOptionTimeOut"; $curl; 10) ]
# Headers include shop ID, language and JSON as content type
Set Variable [ $r ; Value: MBS("CURL.SetOptionHTTPHeader"; $curl; "X-Apple-
SoldTo: " & $soldToCode; "X-Apple-ShipTo: " & $shipToCode;  "X-Operator-User-
ID: " & $userIDCode; "Accept-Language: en_US"; "Content-Type: application/
json"; "Accept: application/json") ]
# For POST, you can add JSON here
If [ Length ( $JSON ) > 0 ]
        Set Variable [ $r ; Value: MBS("CURL.SetOptionPostFields"; $curl; $JSON;
"UTF-8") ]
End If
# Run Transfer
Set Variable [ $r ; Value: MBS("CURL.Perform"; $curl) ]
# Check results.
Set Variable [ $httpResponse ; Value: MBS( "CURL.GetResponseCode"; $curl ) ]
Set Variable [ $result ; Value: MBS("CURL.GetResultAsText"; $curl; "UTF8") ]
Set Variable [ $debug ; Value: MBS("CURL.GetDebugAsText"; $curl; "UTF8") ]
#
If [ $r = "OK" and $httpResponse = 200 ]
        # OK
Else
        # Handle error
End If
Set Variable [ $r ; Value: MBS("CURL.Cleanup"; $curl) ]
```

The script is quite universal as it can be used with different URLs to do various operations. When you pass some JSON, the request becomes a POST request and the JSON is sent. The JSON can be copied from Apple's website and a lot of requests work very nice via MBS Plugin. For example a request with URL "diagnostics/suites?deviceId=" followed by the Device ID (e.g. iPhone's serial number), can query the available diagnostics suites.

We hope you have no problems to implement GSX REST Service into your solution with this help.