

## MYOB Company File: myobAuth

---

#Created: Andrew Duncan - 23 January 2016

#Modified: Andrew Duncan - 07 November 2019

#Purpose: continue MYOB API Authentication (CLOUD)

#Called from button on the Authentication TAB

#Exit script if Desktop API is set

If [ Interface::APIType = "Desktop" ]

Exit Script [ ]

End If

#First check for a code in the <title> tag

Set Variable [ \$htmlContent; Value:GetLayoutObjectAttribute ( "myobAuthWebView" ; "source" ) ]

If [ not PatternCount ( \$htmlContent ; "code=" ) ]

Set Variable [ \$errorText; Value:"The Code Parameter was not received from the MYOB API - please try authenticating again" ]

Beep

Show Custom Dialog [ Title: "Authentication Error"; Message: \$errorText; Default Button: "OK", Commit: "Yes" ]

Exit Script [ Result: \$errorText ]

Else

Set Variable [ \$code; Value:Let ( [  
t1 = GetLayoutObjectAttribute ( "myobAuthWebView" ; "source" ) ;  
n1 = Position ( t1 ; "code=" ; 1 ; 1 ) ;  
n2 = Position ( t1 ; "&scope=" ; 1 ; 1 ) ;

t2 = Middle ( t1 ; n1 + 5 ; n2 - n1 - 5 )

];

t2

) // Let ]

End If

If [ IsEmpty ( \$code ) ]

#code Parameter/variable not received

Set Variable [ \$errorText; Value:"The Code URL Parameter was not received - please try authenticating again" ]

Beep

Show Custom Dialog [ Title: "Authentication Code Error"; Message: \$errorText; Default Button: "OK", Commit: "Yes" ]

Exit Script [ Result: \$errorText ]

End If

#Mandatory Checks

#Check for FileMaker Pro v16 or higher - plug-in not required

If [ \$\$fmVersion < 16 ]

#Call Subscript to check BaseElements plug-in

Perform Script [ "BaseElements Plug-in Checks" ]

Set Variable [ \$pluginCheckResult; Value:Get ( ScriptResult ) ]

If [ not IsEmpty ( \$pluginCheckResult ) ]

Exit Script [ ]

End If

End If

## MYOB Company File: myobAuth

---

**Set Field** [ Interface::zv\_LastAPIResult\_gt; "" ]

*#Branch for plug-in or native steps*

*#Check for FileMaker Pro v16 or higher - plug-in not required*

**If** [ \$\$fmVersion < 16 ]

*#PLUG-IN STEPS*

*#Setup HTTP Headers*

**Set Variable** [ \$Header; Value:BE\_HTTP\_Set\_Custom\_Header ( "Content-Type" ; "application/x-www-form-urlencoded" ) ]

*#Set POST URL*

**Set Variable** [ \$url; Value:MYOB\_AuthURL ]

*#Now set any Proxy values if required*

**If** [ not IsEmpty ( Interface::ProxyServerAddress ) ]

**Set Variable** [ \$setProxy; Value:BE\_HTTP\_Set\_Proxy ( Interface::ProxyServerAddress; Interface::ProxyServerPort ; Interface::ProxyServerUserName ; Interface::ProxyServerPassword ) ]

*// BE\_HTTP\_Set\_Proxy ( proxy {; port ; username ; password } ) ]*

*#Check for error*

**If** [ \$setProxy ≠ 0 ]

**Set Variable** [ \$errorText; Value:"There was an error setting the Proxy Server settings. Plugin error: " & BE\_GetLastError ]

**Beep**

**Show Custom Dialog** [ Title: "Error Setting Proxy"; Message: \$errorText; Default Button: "OK", Commit: "Yes" ]

**Set Field** [ Interface::zv\_LastAPIResult\_gt; \$errorText ]

**Exit Script** [ ]

**End If**

**End If**

**Set Variable** [ \$parameters; Value:"code=" & \$code &

"&redirect\_uri=" & Interface::RedirectURL &

"&client\_id=" & Interface::APIKey &

"&scope=CompanyFile&client\_secret=" & Interface::APISecret &

"&grant\_type=authorization\_code" ]

*#Perform HTTP POST or HTTP PUT*

**Set Variable** [ \$response; Value:BE\_HTTP\_POST ( \$url ; \$parameters ) ]

*#Check Response Code*

**Set Variable** [ \$responseCode; Value:BE\_HTTP\_Response\_Code ]

**Set Variable** [ \$responseHeaders; Value:BE\_HTTP\_Response\_Headers ]

**Set Variable** [ \$error; Value:BE\_JSONPath ( \$response ; "\$.error" ) ]

*#Check for errors*

**If** [ \$response = "?" ]

*#HTTP Response = ?*

**Set Variable** [ \$errorText; Value:"There was an error performing the HTTP POST. Plugin error: " & BE\_GetLastError ]

**Beep**

**Show Custom Dialog** [ Title: "HTTP POST Error"; Message: \$errorText; Default Button: "OK", Commit: "Yes" ]

**Set Field** [ Interface::zv\_LastAPIResult qt; \$errorText ]

**Exit Script** [ ]

**Else If** [ \$responseCode ≠ 200 ]

*#HTTP Response Code < 200*

**Set Variable** [ \$errorText; Value:"There was an error performing the HTTP POST. HTTP Response Code: " & \$responseCode & " Error: " & \$error ]

**Beep**

**Show Custom Dialog** [ Title: "HTTP POST Error"; Message: \$errorText; Default Button: "OK", Commit: "Yes" ]

**Set Field** [ Interface::zv\_LastAPIResult qt; "Response: " & \$response & ¶ &

"Response Code: " & \$responseCode & ¶ &

"Response Header: " & \$responseHeaders ]

**Exit Script** [ ]

**Else**

*#OK - just record the response*

**Set Field** [ Interface::zv\_LastAPIResult qt; \$response ]

**End If**

*#Process HTTP Response*

*#Check Response*

**Set Variable** [ \$accessToken; Value:BE\_JSONPath ( \$response ; "\$.access\_token" ) ]

**Set Variable** [ \$expiresIn; Value:BE\_JSONPath ( \$response ; "\$.expires\_in" ) ]

**Set Variable** [ \$refreshToken; Value:BE\_JSONPath ( \$response ; "\$.refresh\_token" ) ]

*#Set global variables with token details and keys*

**Set Variable** [ \$\$apiKey; Value:Interface::APIKey ]

**Set Variable** [ \$\$apiSecret; Value:Interface::APISecret ]

**Set Variable** [ \$\$myobToken; Value:\$accessToken ]

**Set Variable** [ \$\$refreshToken; Value:\$refreshToken ]

**Set Variable** [ \$\$myobTokenExpires; Value:Get ( CurrentTimestamp ) + \$expiresIn ]

**Set Field** [ Interface::AuthorisationExpires; \$\$myobTokenExpires ]

*#Set global variable to flag already authenticated*

**Set Variable** [ \$\$fmMYOBAuthenticated; Value:1 ]

*#Set a global variable with the MYOB ID for the selected Company File*

**Set Variable** [ \$\$myobCompanyID; Value:SelectedCompanyFile::ID ]

*#Clear out the Web Viewer URL field*

**Set Field** [ Interface::zl\_WebViewerURL qt; "" ]

**Else**

*#NATIVE STEPS - FILEMAKER V16 OR HIGHER*

**Allow User Abort** [ Off ]

*#Reset the debugging global variables*

**Set Variable** [ \$\$cURLTrace; Value:"" ]

*#Set POST URL*

**Set Variable** [ \$url; Value:MYOB\_AuthURL ]

*#Set cURL options*

**Set Variable** [ \$cURL; Value:"--request POST" &  
"--trace \$\$cURLTrace" &

"--header " & Quote( "Content-type" & ":" & "application/x-www-form-urlencoded" ) &

"--data " & Quote( "code=" & \$code ) &  
"--data " & Quote( "redirect\_uri=" & Interface::RedirectURL ) &  
"--data " & Quote( "scope=CompanyFile" ) &  
"--data " & Quote( "client\_id=" & Interface::APIKey ) &  
"--data " & Quote( "client\_secret=" & Interface::APISecret ) &  
"--data " & Quote( "grant\_type=authorization\_code" ) &

"--dump-header \$responseHeaders"

&

Case (

not IsEmpty ( Interface::ProxyServerAddress ) ;

"--proxy " & Interface::ProxyServerAddress & ":" & Interface::ProxyServerPort &  
"--proxy-user " & Interface::ProxyServerUserName & ":" & Interface::ProxyServerPassword

) ]

*#Perform HTTP REQUEST*

**Set Error Capture** [ On ]

**Insert from URL** [ \$response; \$url; cURL options: \$cURL ]  
[ Select; No dialog ]

**Set Variable** [ \$lastError; Value:Get (LastError) ]

**If** [ \$lastError ≠ 0 ]

*#Insert from URL Error*

**Set Variable** [ \$errorText; Value:"There was an error performing the Insert from URL HTTP Request. Error: " & \$lastError  
& " (" & ErrorDescription ( \$lastError ) & ")" ]

**Beep**

**Show Custom Dialog** [ Title: "HTTP Request Error"; Message: \$errorText; Default Button: "OK", Commit: "Yes" ]

**Set Field** [ Interface::zv\_LastAPIResult \_gt; \$errorText ]

**Exit Script** [ Result: \$errorText ]

**End If**

*#Check Response Code*

**Set Variable** [ \$responseCode; Value:httpResponseCode ( \$responseHeaders ) ]

*#Check for Errors*

**Set Variable** [ \$error; Value:JSONGetElement ( \$response ; "error" ) ]

**If** [ \$responseCode ≠ 200 ]

*#HTTP Response Code <> 200*

**Set Variable** [ \$errorText; Value:"There was an error performing the HTTP Request. HTTP Response Code: " &  
\$responseCode & " Error: " & \$error & " " & \$errorAdditionalDetails ]

**Beep**

**Show Custom Dialog** [ Title: "HTTP Request Error"; Message: \$errorText; Default Button: "OK", Commit: "Yes" ]

**Set Field** [ Interface::zv\_LastAPIResult \_gt; \$errorText ]

**Exit Script** [ Result: \$errorText ]

**Else**

*#OK - just record the response*

**Set Variable** [ \$response; Value:JSONFormatElements ( \$response ) ]

**Set Field** [ Interface::zv\_LastAPIResult\_gt; \$response ]

**End If**

*#Process HTTP Response*

*#Check Response*

**Set Variable** [ \$accessToken; Value:JSONGetElement ( \$response ; "access\_token" ) ]

**Set Variable** [ \$expiresIn; Value:JSONGetElement ( \$response ; "expires\_in" ) ]

**Set Variable** [ \$refreshToken; Value:JSONGetElement ( \$response ; "refresh\_token" ) ]

*#Set global variables with token details and keys*

**Set Variable** [ \$\$apiKey; Value:Interface::APIKey ]

**Set Variable** [ \$\$apiSecret; Value:Interface::APISecret ]

**Set Variable** [ \$\$myobToken; Value:\$accessToken ]

**Set Variable** [ \$\$refreshToken; Value:\$refreshToken ]

**Set Variable** [ \$\$myobTokenExpires; Value:Get ( CurrentTimestamp ) + \$expiresIn ]

**Set Field** [ Interface::AuthorisationExpires; \$\$myobTokenExpires ]

*#Set global variable to flag already authenticated*

**Set Variable** [ \$\$fmMYOBAuthenticated; Value:1 ]

*#Set a global variable with the MYOB ID for the selected Company File*

**Set Variable** [ \$\$myobCompanyID; Value:SelectedCompanyFile::ID ]

*#Clear out the Web Viewer URL field*

**Set Field** [ Interface::zl\_WebViewerURL\_gt; "" ]

*#End \$\$fmVersion < 16 test*

**End If**

**Commit Records/Requests**

*#Show dialog*

**Beep**

**Show Custom Dialog** [ Title: "Authentication Succesful"; Message: "You have successfully authenticated with MYOB"; Default Button: "OK", Commit: "Yes" ]

**Exit Script** [ ]

*#History:*

*#v1.00: Andrew Duncan - 23 January 2016*

*#v1.01: Andrew Duncan - 07 November 2019*

*#- updated for native FileMaker Pro v16 or higher functions/scripts*