

# Translate PHP script with CURL to FileMaker Script

Today we want to show you how to translate a PHP script with CURL commands step by step to a FileMaker script. Here is a PHP script:

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_SSL_VERIFYPeer, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
curl_setopt($ch, CURLOPT_POST, TRUE);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS, $params);
curl_setopt($ch, CURLOPT_HTTPHEADER, array(
'Content-Type: application/json',
'Content-Length: ' . strlen($params))
);
$result = curl_exec($ch);
curl_close($ch);
```

So lets go through the script line by line. First line is "\$ch = curl\_init();" which simply creates a new curl session in PHP. The result is the handle which is stored in \$ch variable. This handle value (a number) is than used to to reference the CURL session later in the script. In FileMaker, the call to the MBS plugin is MBS("CURL.New"). This function starts a new session and we store it in a variable, exactly as in PHP.

Next calls set a couple of options. First for SSL, you should use a certificate to have the plugin verify the identity of the server. But in this example we simply disable verification. So we use the calls MBS("CURL.SetOptionSSLVerifyPeer"; \$curl; 0) and MBS("CURL.SetOptionSSLVerifyHost"; \$curl; 0). The option CURLOPT\_RETURNTRANSFER in PHP is not needed by the plugin. The plugin automatically collects the data downloaded and your script can query the result later.

Next option sets the URL to use. This can be an URL for one of the supported protocols including FTP(S), HTTP(S) or SFTP. In the script we use MBS("CURL.SetOptionURL"; \$curl; \$URL) to pass the URL as text to the plugin. Than we can set other options like a 30 seconds timeout using MBS("CURL.SetOptionTimeout"; \$curl; 30).

As this example transfer is a HTTP Post operation for sending form data, we have to turn on the post option. All the web services like SOAP normally use POST queries which use the request xml as the form data. So with the call MBS("curl\_setopt"; \$curl; 1) we enable post operation. With the custom request option you can of course explicit also set POST, but here it is optional: MBS("curl\_setopt"; \$curl; "POST"). Often we custom requests like DELETE with a REST API.

For the actual content of the transfer we have a couple of functions to set input data. So for the CURLOPT\_POSTFIELDS we can use the plugin function MBS("curl\_setoptPostFields"; \$curl; \$params) with the right text in the variable. The plugin will copy the data for the transfer and also set internally the size for the transfer. So when setting the http headers (CURLOPT\_POSTFIELDS), we don't need to include the Content-Length line with MBS Plugin. For the http headers we need to pass the content type to the plugin to inform our web server that we pass data as JSON here: MBS("curl\_setoptHTTPHeader"; \$curl; "Content-Type: application/json").

To run the transfer, the PHP script calls curl\_exec function. In the plugin we call MBS("curl\_perform"; \$curl) and get back an error code. This error code is zero on success. For a list of error codes, please check the plugin documentation. The PHP function returns the result directly. But with our plugin, we have the function CURL.GetResultAsText. Depending of the transfer, the result could be text, an image or some other data. To show text in a field in FileMaker, we also need to make sure the line endings match. FileMaker always uses Mac line endings, so we use the String.ReplaceNewline function. So we get the result with the calculation MBS("String.ReplaceNewline"; MBS("curl\_getresultasText"; \$curl);1).

To know what reason a failure has, you can get debug messages. So before the perform call, we enable verbose debug messages with a call to MBS("curl\_setoptVerbose"; \$curl; 1). To actually get the text for showing in a field, we query the messages as text with this call: MBS("String.ReplaceNewline"; MBS("curl\_getDebugAsText"; \$curl);1).

Finally we do some cleanup. In PHP the curl\_close call and in FileMaker the MBS("curl\_cleanup"; \$curl) call. As you see the function calls are quite similar and you can easily translate PHP scripts using CURL for use in FileMaker.

The final script looks like this:

```
Set Variable [$curl; Value: MBS("CURL.New")]
Set Variable [$result; Value: MBS("CURL.SetOptionURL"; $curl; $URL)]
Set Variable [$result; Value: MBS("CURL.SetOptionSSLVerifyPeer"; $curl; 0)]
Set Variable [$result; Value: MBS("CURL.SetOptionSSLVerifyHost"; $curl; 0)]
Set Variable [$result; Value: MBS("CURL.SetOptionVerbose"; $curl; 1)]
Set Variable [$result; Value: MBS("CURL.SetOptionTimeout"; $curl; 30)]
Set Variable [$result; Value: MBS("CURL.SetOptionPost"; $curl; 1)]
Set Variable [$result; Value: MBS("CURL.SetOptionPostFields"; $curl;
$params)]
Set Variable [$result; Value: MBS("CURL.SetOptionHTTPHeader"; $curl;
"Content-Type: application/json")]
Set Field [Webservice::Error; MBS("CURL.Perform"; $curl)]
Set Field [Webservice::Input; $input]
Set Field [Webservice::Result; MBS("String.ReplaceNewline";
MBS("CURL.GetResultAsText"; $curl);1)]
Set Field [Webservice::DebugMessages; MBS("String.ReplaceNewline";
MBS("CURL.GetDebugAsText"; $curl);1)]
Set Variable [$result; Value: MBS("CURL.Cleanup"; $curl)]
```