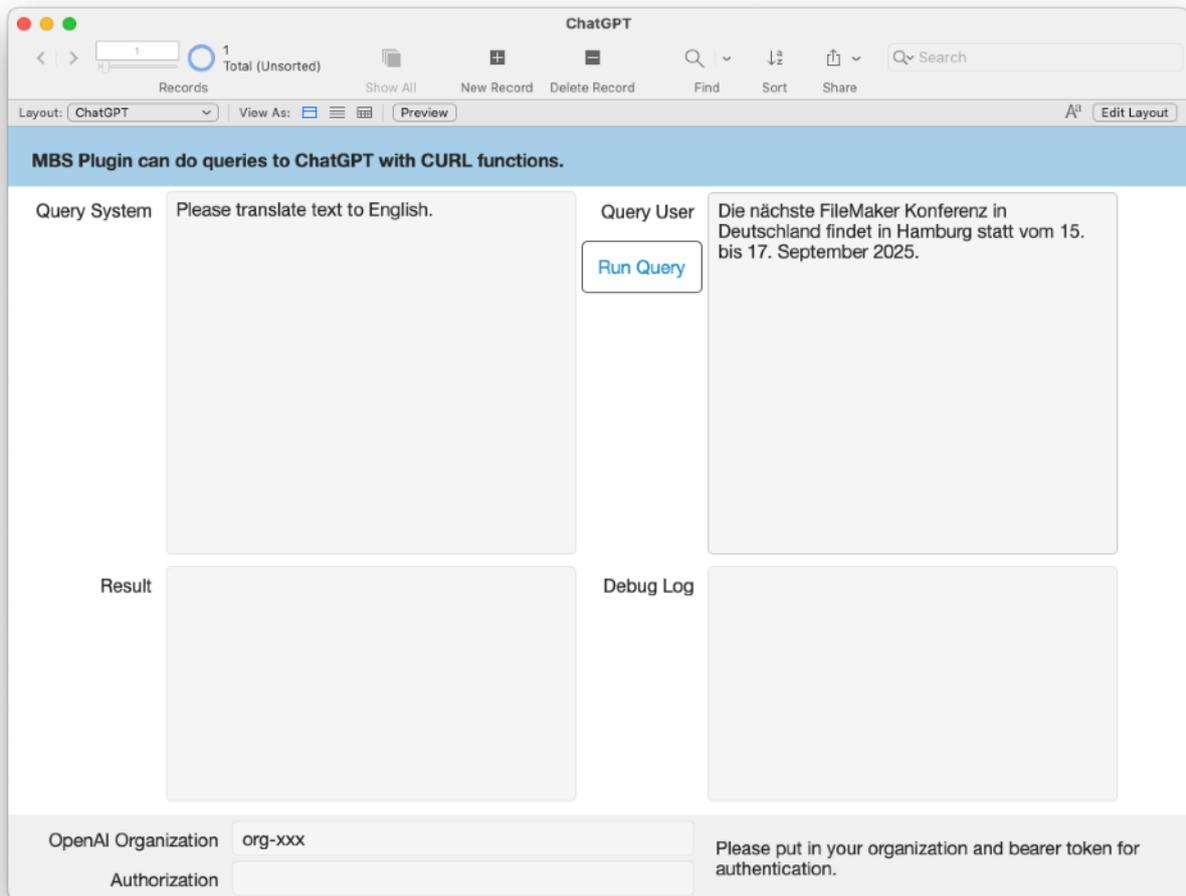


# Using ChatGPT in FileMaker

From time to time we get asked to provide an example for ChatGPT. Since this is just another web service, we can just handle it with the CURL functions in MBS Plugin. But since we don't like to block the user interface while ChatGPT processes the request, we use our CURL.PerformInBackground function to run it in the background. Later when finished, it triggers a second script to process the result.



The request is build with a call to JSONSetElement here passing the various fields. We include a system message with the request, e.g. "Please translate text to English." and then pass the text to translate in the user role. This way the user should not be able to provide instructions to the LLM in their text.

To test the script, please get an account for OpenAI. In the headers we need to pass your organization ID from and the bearer token you got there. Each query will use some tokens, but we limit the answer to 100 tokens.

Here is the full script to start the query:

```
Set Variable [ $curl ; Value: MBS("CURL.New") ]
Set Variable [ $json ; Value: JSONSetElement ( "{}" ;
    ["model" ; "gpt-4o-mini" ; JSONString];
    ["max_tokens" ; 100 ; JSONNumber];
    ["messages.[0].role" ; "system" ; JSONString];
    ["messages.[0].content" ; ChatGPT::Query System ; JSONString];
    ["messages.[1].role" ; "user" ; JSONString];
    ["messages.[1].content" ; ChatGPT::Query User ; JSONString] ) ]
Set Variable [ $result ; Value: MBS("CURL.SetOptionPostFields" ; $curl ; $json) ]
Set Variable [ $result ; Value: MBS( "CURL.SetOptionHTTPHeader" ; $curl ;
    "OpenAI-Organization: " & ChatGPT::OpenAI Organization ;
    "Authorization: Bearer " & ChatGPT::Authorization ;
    "Content-Type: application/json" ) ]
Set Variable [ $result ; Value: MBS("CURL.SetOptionURL" ; $curl ; "https://
api.openai.com/v1/chat/completions" ) ]
Set Variable [ $result ; Value: MBS("CURL.SetFinishedScript" ; $curl ;
Get(FileName) ; "Finished" ) ]
Set Variable [ $result ; Value: MBS("CURL.PerformInBackground" ; $curl) ]
```

Here is the finished script, which gets called when the transfer is done. The script parameter is the CURL session reference number. We query debug messages and the result as JSON. We can then pick the output message from the JSON and show it in the field:

```
Set Variable [ $curl ; Value: Get(ScriptParameter) ]
Set Variable [ $debug ; Value: MBS("CURL.GetDebugMessages" ; $curl ;
"UTF-8" ) ]
Set Variable [ $result ; Value: MBS("CURL.GetResultAsText" ; $curl ; "UTF-8" ) ]
Set Field [ ChatGPT::Debug Log ; $debug ]
Set Field [ ChatGPT::Result ; $result ]
#
Set Variable [ $text ; Value: JSONGetElement ( ChatGPT::Result ;
"choices[0].message.content" ) ]
If [ Length ( $text ) > 0 ]
    Set Field [ ChatGPT::Result ; $text ]
End If
#
Set Variable [ $r ; Value: MBS("CURL.Release" ; $curl) ]
```

Please try and see whether this can help you add ChatGPT to the projects. Translation is a good thing and you can even specify whether you like formal or informal text.

We will include the example file with the next plugin for you to use.