

Query geolocation in FileMaker via MBS Plugin and Google Maps API

You may know that we have CLGeocoder functions in our plugins to query geo coordinates for addresses on Mac. But some users need it cross platform and I recently implemented a sample script for an user.

So first get a Google Account. It's always good to separate this, so best may be to make a new account for a project where you need maps. Best on the client, so Google knows where to complain about abuse.

Then go to the [developer page for the google maps api](#). There you can [request a new API key](#). Please copy the API key into the field of my example database. Or just hard code it in the script. Although for client projects, it's always good to have a preferences layout where you can set global settings like this and you don't need to check all scripts whether they use the key.



The script starts a CURL session, encodes the address to query into the URL and runs the query. The JSON is put in a field and the interesting values are extracted: Latitude, longitude and formatted address. Please check the script:

#Start new session

```
Set Variable [$curl; Value:MBS("CURL.New")]
```

#Set URL to load (HTTP, HTTPS, FTP, FTPS, SFTP, etc.)

```
Set Variable [$result; Value:MBS("CURL.SetOptionURL"; $curl; "https://  
maps.googleapis.com/maps/api/geocode/json?address=" & MBS("Text.EncodeToURL";  
Substitute(Google Maps API::Address to query; ¶; ", "); "utf8") & "&key=" & Google Maps  
API::API Key)]
```

#RUN now

```
Set Field [Google Maps API::Result; MBS("CURL.Perform"; $curl)]
```

#Check result

```
Set Field [Google Maps API::JSON; MBS("CURL.GetResultAsText"; $curl; "UTF8")]
```

```
Set Field [Google Maps API::Debug Messages; MBS("CURL.GetDebugAsText"; $curl)]
```

```
Set Variable [$httpResult; Value:MBS("CURL.GetResponseCode"; $curl)]
```

```
Set Variable [$status; Value:MBS("JSON.GetPathItem"; Google Maps API::JSON;  
"status"; 1 )]
```

```
If [$status = "OK"]
```

```
    Set Field [Google Maps API::Latitude; MBS("JSON.GetPathItem"; Google Maps  
API::JSON; "results¶0¶geometry¶location¶lat"; 1 )]
```

```
    Set Field [Google Maps API::Longitude; MBS("JSON.GetPathItem"; Google Maps  
API::JSON; "results¶0¶geometry¶location¶lng"; 1 )]
```

```
    Set Field [Google Maps API::Found; MBS("JSON.GetPathItem"; Google Maps  
API::JSON; "results¶0¶formatted_address"; 1 )]
```

```
Else
```

```
    Set Field [Google Maps API::Latitude; ""]
```

```
    Set Field [Google Maps API::Longitude; ""]
```

```
    Set Field [Google Maps API::Found; ""]
```

```
End If
```

#Cleanup

```
Set Variable [$result; Value:MBS("CURL.Cleanup"; $curl)]
```

I hope this helps you. The example database will be in the next plugin release. Or email me if you need a copy today.