

Xojo Web Apps on FileMaker Cloud Server

Your client or you yourself may have a FileMaker Cloud server. Beside using it as a way to host FileMaker solutions you can actually do more with that server.



Data Files on FileMaker Cloud

Did you know you can put data files to be delivered by Apache web server into the /FileMakerData/HTTPDocs/httpsRoot folder?

You may need to login via ssh and set permissions on the folder to be writeable for you. Just inside the httpsRoot folder, you can upload any file you like to offer for download:

/FileMakerData/HTTPDocs/httpsRoot

In this folder you can upload PDF documents or video files and then link them in emails or a WebViewer.

As an example, you can use MBS Plugin functions in a script to write files into this folder:

```
MBS("Text.WriteTextFile"; "/FileMakerData/HTTPDocs/httpsRoot/test.txt"; "UTF-8")
```

Or

```
MBS( "Container.WriteFile"; "/FileMakerData/HTTPDocs/httpsRoot/test.pdf"; MyTable::MyContainer )
```

Now the files there are distributed by Apache Server to all users with the right URL, e.g.

<https://yourname.filemaker-cloud.com/test.jpg>

This can greatly reduce the load on your FileMaker server for distributing files as well as allow you to provide files to users without using a FileMaker connection. If needed your solution can also write an .htaccess file to provide a simple password protection to a folder.

Xojo Console Apps

You can build Xojo console apps for Linux 64-bit, upload them to the FileMaker Cloud Server and just run it via ssh. That is no problem and no additional libraries are needed, unless you use a declare or a plugin with dependencies. For example some MBS Plugins for LDAP or SQL connections need additional libraries.

Xojo Web App as Standalone

You can build Xojo Web Apps and run them on these servers in addition to your FileMaker Web Direct solution.

A stand alone server can run there and use a higher port number. To allow the service, you need to login and run the app as daemon, e.g. with a & operator. See Xojo's deployment documentation for linux for details. For your AWS server, please check security group settings (external firewall) and add a rule to allow the port. Also login to your server via ssh and add a rule to the internal firewall:

```
sudo firewall-cmd --permanent --zone=public --add-port=9001/tcp
sudo firewall-cmd --reload
```

This example opens port 9001. For me port 9000 is blocked by some other service.

Now you can use your web app with an URL like `http://yourname.filemaker-cloud.com:9001/` in a browser.

Xojo Web App with CGI

While stand alone can be more responsive by using web sockets and skipping apache + cgi as extra layers, you may prefer to use apache. The apache Webserver will do the SSL encryption and the CGI script will automatically launch the Xojo app on demand.

First, you need to install perl cgi module:

```
sudo yum install perl-CGI
```

And edit `httpd.conf` in `/FileMakerData/HTTPConf` to add another load module call:

```
LoadModule cgi_module /usr/lib64/httpd/modules/mod_cgi.so
```

Further down, you need to uncomment the `AddHandler CGI` line:

```
AddHandler cgi-script .cgi
```

Further down, we also need to allow `ExecCGI` for the folder, so you add this directory configuration:

```
<Directory "${HTTP_ROOT}/htdocs/httpsRoot">
Options +ExecCGI +FollowSymLinks +MultiViews
AllowOverride All
</Directory>
```

Restart server here to apply changes. I didn't figure out the command to restart just apache, but if you know, please post a comment.

Finally you can upload your Xojo web app to a subfolder in `/FileMakerData/HTTPDocs/httpsRoot/` and start it. Permissions must be correct: directory writeable, `config.cfg` writable and `cgi` and app itself be executable. But otherwise it should just work as like any other web deployment.

Future

By having both FileMaker and Xojo apps on the same server, allows us to intermix both for building solutions. Using one server for both instead of two is a great cost saving. FileMaker 16 platform allows rapid database app design and provides REST Data interface as well as functions based on CURL to query services. With Xojo we can build custom services to run on the same machine and do things like pushing data from/to FileMaker. We'll see what great solutions can be build by combining Xojo with FileMaker here in the future.

PS: Please note that you put files in folder `"/FileMakerData/HTTPDocs"`, but your app needs to access the files using the `"/opt/FileMaker/FileMaker Server/HTTPServer/htdocs/"` path.