Gisaf - Support #15439

Migration to "Gisaf NG"

21/04/2024 12:05 - Philippe May

Status:	In Progress	**Start date:**	21/04/2024
Priority:	Normal	**Due date:**	
Assignee:	Philippe May	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour

Description

I have installed last night (20/4/2024) "the" big update for Gisaf, that i worked on these past few months. We can call it "Gisaf NG" (New Generation) in our exchanges.

I checked that most of Gisaf runs as usual (map, dashboards, measures, admin). But there are quite a few things that i still need to fix, some of them i know, and probably others that i haven't noticed yet.

In the next few days i'll fix those that i already noticed - please bear with me as i'll be traveling a bit. You can add the issues that you notice here.

Some technical details

There are 2 main updates in the underlying code that Gisaf is made of.

FastAPI

See https://fastapi.tiangolo.com/. It does the main job of handling HTTP requests following a clean standard called OpenAPI (see https://www.openapis.org/).

This replaces GraphQL (https://graphql.org/), another standard that was used by the "old" Gisaf. OpenAPI is much easier to deal with in general, lightweight, and can integrate more easily with other tools.

FastAPI provides a handy tool for exploring the functions provided by the server side of Gisaf: see https://gis.avcsr.org/docs .

Because of that radical change on the server side, i had to completely revamp the client, migrating all GraphQL calls to OpenAPI ones. This usually does not change anything to the user experience.

One notable exception is the "login" (or logout), that now pops up a dialog, Once logged in (or logged out), the user interface shows (or hides) the functions that the user is allowed to perform, without the need to refresh the page.

It's worth noting that there's another small component that greatly simplifies the development of the client, thanks to OpenAPI: https://openapi-ts.pages.dev/introduction .

SQLModel

See https://sqlmodel.tiangolo.com/, This was the initial motivation for that big update, as this allows Gisaf to drop Gino (https://python-gino.org/), a core library which is clearly not maintained anymore.

SQLModel interfaces with the database in a very neat way, using new Python capabilities that were not available when Gisaf was initially developed. Beside, SQLModel is really well integrated with FastAPI.

The result is a much more straightforward integration between the database, the server, the API interface and the final user interface.

Hosting the code

The new version of Gisaf is now split in 2 separate new repositories (client and server). There are hosted in a public portal:

https://codeberg.org/Philo/Gisaf
https://codeberg.org/Philo/gisaf-app

For CSR Geomatics (models and extensions), there are 2 more repositories:

https://codeberg.org/Philo/avgs_models
https://codeberg.org/Philo/avgs_device_interfaces

History

#1 - 21/04/2024 12:21 - Philippe May

- Description updated

#2 - 29/04/2024 03:44 - Philippe May

Download of layers (gpkg and DXF) fixed