GeoMapFish User-Group

GeoMapFish

March 11th 2021

Agenda

1. General information
2. A little more autonomy with GMF
    1. Improve documentation
    2. Getting started
3. Future of GMF
    1. Strategies for replacing AngularJS
    2. Reflections in progress
4. Tour de table / Meinungsrunde
5. GMF version 2.6
    1. Current state
    2. Demo

1. General Information

Contact Email

contact@geomapfish.org

User-Group Email

geomapfish-users@googlegroups.com

Feature Requests

https://github.com/camptocamp/GeoMapFish

Website

https://geomapfish.org

Contribute

https://github.com/geomapfish/geomapfish_website

2. More autonomy for users

Improve documentation

Components to improve:

- nGeo
- c2cgeoportal
- **Admin interface**
- Yaml files
- **Migration documentation (changelog)**
- Getting started

Improve documentation

- Admin interface
  - Contextual documentation
  - Better documentation for metadata and functionalities
  - Options :
    - Filter metadata and functionalities accord to context
    - Backport GMF 2.4 / 2.5
- Migration documentation (changelog)
  - Document important changes between version (from version 2.4 - docker)
  - Exhaustive list of all TODOs when migrating

Improve documentation - Offer

FEATURE	DEV Hours	TESTING Hours	PROJECT MNGMT Hours	COST CHF excl. taxes
Admin documentation	61.2		9.2	10'560
Change log	13.2		2.0	2'280
Options				
Metadata and functionnalities filter	12.5	3.8	2.4	2'805
Admin documentation backport to 2.4	28.0		4.2	4'830
Total without options	**74.4**	**0.0**	**11.2**	**12'840**
Total	**114.9**	**3.8**	**17.8**	**20'475**

Website

GeoMapFish Home Functionalities Documentation Gallery Community Roadmap

Documentation

Lang de fr

GeoMapFish

Master version

Documentation:

- Client part: ngeo
- Server part: c2cgeoportal

Examples:

- GeoMapFish
- ngeo

LTR version (2.4)

Documentation:

- Client part: ngeo
- Server part: c2cgeoportal

Examples:

- GeoMapFish
- ngeo

Other applications

Web map server:

- MapServer
- QGIS Server

Web map client:

- OpenLayers

Print server:

- MapFish Print

Getting Started

- The “Getting Started” documentation won’t be done by Camptocamp.
- The PSC members have written a first version: https://github.com/geomapfish/getting_started
- We’re counting on the GeoMapFish community to make it grow.

3. The future of GeoMapFish

Strategies for replacing AngularJS

Requirements:

- Keep all the existing functionalities
- Same level of ergonomy and performance
- No breaking change neither in the backend, nor in the administration part
- Ideally 10 years support on the new framework

Strategies for replacing AngularJS

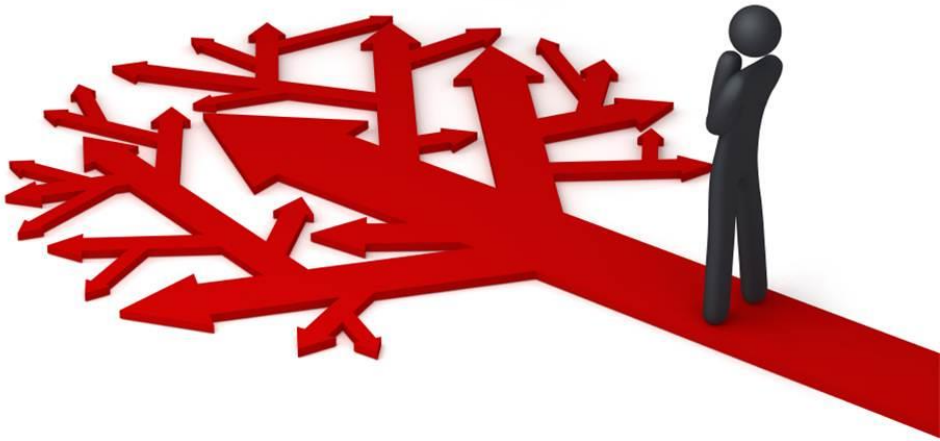
Evaluation criteria:

- Cost
- Migration effort from older versions
- Low learning curve
- Community
- Durability

Strategies for replacing AngularJS

4 different scenarios:

1. Stay on AngularJS with extended support
2. Rough 1:1 migration
3. Progressive migration
4. Switch to another existing library



1. Stay on AngularJS with extended support

Pros:

- More time to consider others solutions
- Wait for a opportunity to migrate
- No short-term cost

Cons:

- Keep developing in a deprecated technology
- Only postpones the migration
- Likely no expansion of the community

Horizon:

- Now

2. Rough 1 to 1 migration

Pros:

- Capitalize on existing code
- Probably the less expensive migration.
- Could be externalized.

Cons:

- No improvement of GMF during migration
- The first results will only be really visible once the migration is almost complete.
- Some components may not be able to be migrated as is
- We remain closely linked to the new chosen framework
- In 7-10 years, the same question of a technical migration will arise again

Horizon:

- 1-2 years

3. Progressive migration

Pros:

- Capitalize on existing code
- GMF can get new functionalities during the migration
- We can choose component by component the best way to migrate it
- Possibility to develop some parts independently of a framework, in pure javascript

Cons:

- Probably higher costs
- During the migration, old and new technologies may have to cohabit, and this can add complexity to the code and in the architecture
- Today, no solution of cohabitation between AngularJS and another Framework has been tested ==> we first need a Proof of Concept to check the feasibility.

Horizon:

- 2-4 years

4. Switch to another existing library

Pros:

- Join an existing community
- Other funding opportunities

Cons:

- Almost zero capitalization on the existing GMF
- The server part need most likely to be rewritten
- Integration of the existing configuration (layer tree, metadata, …) will be very difficult
- Customization will be very difficult
- Dependency on another community, possible difficulties to collaborate with it
- Some components may not be able to be migrated in the new library as is
- Cost are hard to evaluate

Horizon:

- 2-3 years ?

Reflections in progress

1. ~~Stay on AngularJS with extended support~~
2. Rough 1:1 migration
3. Progressive migration
4. ~~Switch to another existing library~~

Estimated costs

For a 1 to 1 migration : **about 250 000 CHF**

- Preliminary refactoring done by C2C
- Outsourced migration to an US company specialized in AngularJS/Angular

  https://xlts.dev
- PM & Tests done by Camptocamp and the PSC

For a progressive migration : **not estimated yet, but very probably higher**

For reference:

- User-Group investment in 2019 : 136 000 CHF
- User-Group investment in 2020 : 134 000 CHF

Reflections in progress

Step 1 : Release 2.7 of GMF

- Proof of Concept
- Merge of ngeo/gmf components
- Remove dependencies to AngularJS where it is easily done
- Ensure compatibility with standard Web Components
- Pure technical migration (no new functionality)
- LTS Release early 2022

What are Web Components ?

Developers have long been able to define reusable web elements using JavaScript frameworks such as React and Angular. However, each framework uses a different standard which often prevents useful snippets of code from being used across different projects. This problem can be overcome by using web components which are reusable HTML components that can be used regardless of the framework. Standardized in 2012, the web element type is now supported by all current browsers.

https://www.ionos.com/digitalguide/websites/web-development/web-components/

Reflections in progress

Goal of this first step:

- Technical choices will be validated
- The complexity of the client code will have been simplified
- Give users time to prepare for the migration
- Custom developments can be migrated to a technology independent of a framework

Reflections in progress

Step 2 :

Depending on the results of the first step, and depending on the finances, there will be 2 options.

Reflections in progress

Option 1 :

1 to 1 migration of the remaining components and of the core of GMF

- Replace completely AngularJS with a new framework (Angular, VueJs, React, …)
- Mostly a technical migration
- Possible integration of some new features (but not sure)
- Release GMF 3.0 in 2023 without AngularJS any more
- The new version is linked to a new framework

Reflections in progress

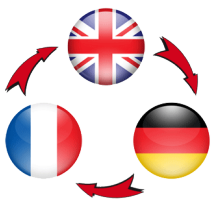
Option 2 :

Start a gradual migration of the remaining components

- Convert step by step the GMF components into standard Web Components
- Integration of some new features
- Keep AngularJS a little longer
- Release GMF 2.8 in 2023 with new functionalities and a little less AngularJS
- Release GMF 2.9 in 2024 with new functionalities and even a little less AngularJS
- Release GMF 3.0 in 2025 without AngularJS any more
- The new version uses the new framework as few as possible

4. Tour de table / Meinungsrunde

- Who are you?
- What is your current situation with GMF?
- What are your expectations / fears for the next years?



- Is the PSC going in the right direction regarding documentation?
- What do you think about the migration scenarios?
- How do you plan to participate financially in this migration?

5. GMF Version 2.6

Current state : developments done

Number	Category	Feature	Description
2.6.16	Admin interface	Admin lists	Add layer list in restriction area
2.6.20	Editing	Editing client-side type validation	When editing an object and depending on the browser, verify the data type
2.6.22	Editing	Editing dropdown lists sorting	Add a column named "order" to sort dropdown lists.
2.6.54	Editing	Overlapping geometries	Allow editing of overlayed geometries in edit and redline interface
2.6.34	External layers	Legend of imported WMS layers	Display legend of imported WMS layers
2.6.35	External layers	External WMS GetFeatureInfo	Simple interrogation of external WMS (Geoadmin) using WMS getfeatureinfo
2.6.47	External layers	Drag/drop files	Instead of using external entry plugin, permit drag and drop of files directly on the map
2.6.24	Login	Server logout message	Push a message to user when server authentication is lost and client interface shows that user is connected.
2.6.57	Mobile	Auto geolocation mobile	Geolocation activated at startup
2.6.29	Print	Print legend improvments	Add groups in legends
2.6.10	Query	CSV export in Window	New button next to results filter. Exports everything (no slection like with grid)
2.6.38	Query	Message when too many results	Add message when query count is too big
2.6.46	Query	Click query tolerance per layer	When doing a click query, be able to define a tolerance on each layer
2.6.53	Query	Scale limit on WMTS query	Scale limit on WMTS query
2.6.1	Redlining	Redlining arrows	Allow to draw an arrow
2.6.31	Redlining	Snapping for measures/drawing	Snapping functionality for measuring and drawing tools
2.6.26	Search	Search keywords for layers	Possiblitiy to add keywords to layers in order to find them easier in the search
2.6.27	Search	Full-text documentation	Improve Full-text-search documentation
2.6.25	Shortlink	Shortlink with time	Include time configuration on layers in shorlinks
2.6.32	User interface	Resizable right panel	Right panel resizable for all tools. Improve GUI.
2.6.55		GetCapabilities sync	
2.6.56		Mapillary integration	

Current state : integration tests have started

Planned release date : end of April 2021

2. Demo


thank you
danke
merci
gracias
obrigado
dziękuję
bedankt
teşekkür ederim
go raibh maith agat
sukriya
kop khun krap
terima kasih
감사합니다
ευχαριστώ
спасибо
рахмат
謝謝
ngiyabonga
شكراً جزيلاً
Баярлалаа
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
хвала
asante
manana
obrigada
tenki
murakoze
chokrane
mamnun
дякую
djiere dieuf
tau
mochchakkeram
kiitos
dankie
nanni
nandri
dhanyavad
hvala
mauruuru
köszönöm
akun
dankon
aciu
gràcie
bayarlalaa
enkosi
chnorakaloutioun
gratias ago
gracies
sulpáy
taiku
sobodi
dèkuji
mèsi
sagolun
didi madloba
najis tuke
kam sah hamnida
rahmat
তোমাকে ধন্যবাদ
ありがとう
tanemirt
rahmet
grazie
arigatô
takk
dakujem
trugarez
merce
мерси
shukriya
diolch
dhanyavadagalu
xièxie