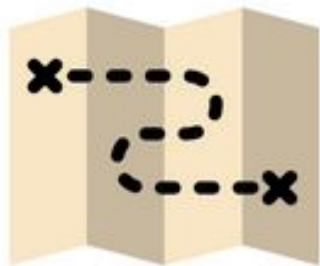


Improving Grafana plugins

Claire VELUT – Mica MURPHY



Grafana



Table of content

Introduction	3
Context and aim	3
Plugins	4
TrackMap	4
QRCode	4
Technologies	5
Grafana	5
InfluxDB	5
JavaScript	5
AngularJS	5
React	5
Conclusion	6
To go further	6
Trackmap	6
QRCode	6
Get perspective	6

Introduction

The purpose of this document is to present the work achieved during the project of the fourth year of Computer Science at Polytech Grenoble. Alongside the other documents and the project repository, it helps to understand the project in its entirety.

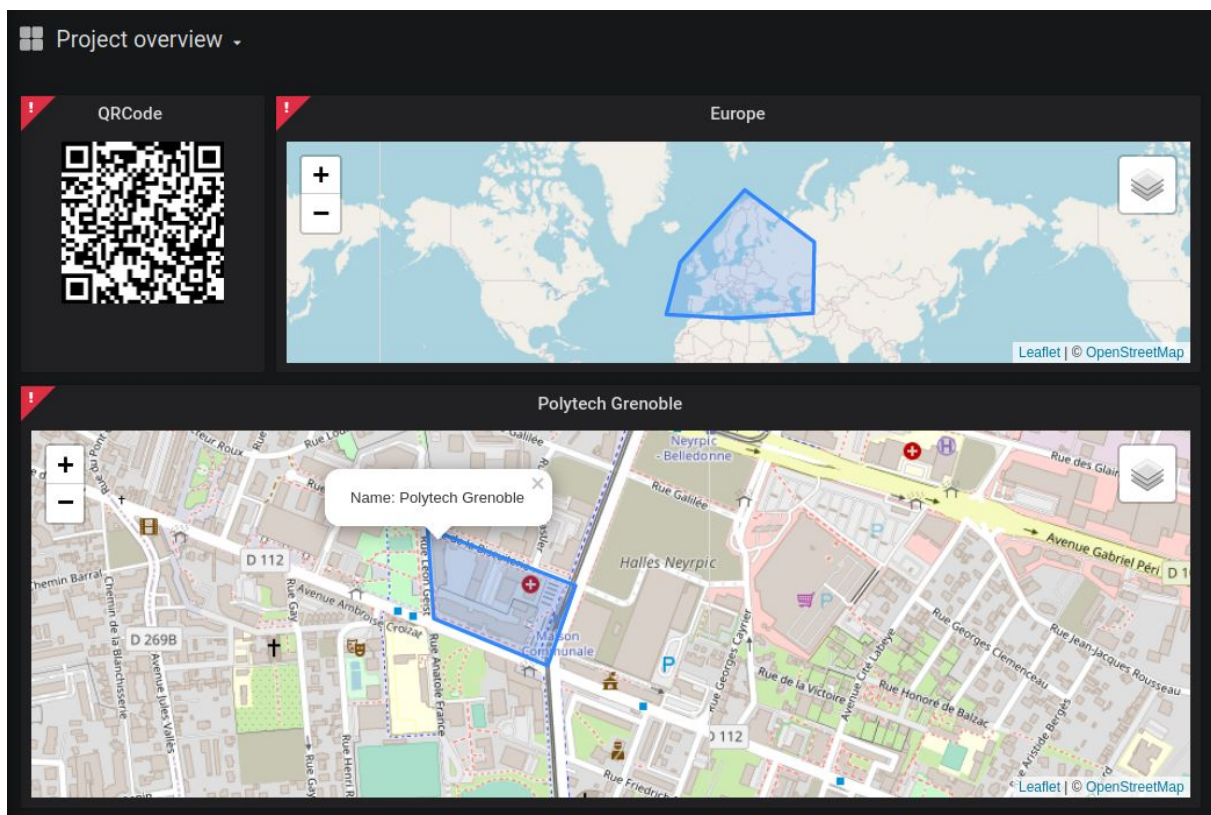
We will present the context and aim of the project. We will describe the technologies we have used, the choice we have made, the difficulties we have encountered. And we will finish with ideas to pursue this project.

You can find all the documentation and code available via this link:

<https://gricad-gitlab.univ-grenoble-alpes.fr/Projets-INFO4/19-20/12>

Context and aim

Grafana is an open source software that allows to visualize and analyze metrics data through dashboards. It is used in lots of Polytech project and on the campus. Our project goal was to improve grafana plugins and to contribute the modifications to the grafana community. The first part was to improve an already existing grafana plugin, TrackMap. The second part was to create a panel that displays a QR code of the current dashboard.



An example dashboard with panels using the Trackmap and QRCode plugins

Plugins

TrackMap

The Trackmap plugin allows to visualizes GPS points as a line on an interactive map based on an time series database like InfluxDB. We were asked to add the possibility to add GeoJSON data to the map by copy/pasting the data in a text area or by importing a file. We started by reading and understanding the code, then searching where we needed to edit or add code. The two main files we modified are `src/trackmp_ctrl.js` (the controller) and `src/partials/module.html` (the view).

As we coded, we were getting used to this plugin and how to contribute to it, so we chose to continue improving this panel with our own ideas. We first added the possibility to save an overlay through a text area in which a GeoJSON file is pasted. We then add a list that displays the different overlays added and the possibility to delete or to download an overlay of the list.

One problem we encountered was pretty problematic: we could create a Trackmap panel, edit it and see the modifications appear on the preview map but we could not save the dashboard. It was complicated to find what caused this "Unexpected Error" and we went back one by one commit in order to locate the source. We did small commits so even if it was long it was not too hard to do. The error occured because not all the objects could be serialized correctly with AngularJS and JavaScript (in particular the Leaflet objects). The workaround we found was to use more lists, one that is saved and others that don't need to be saved because they can be created again from the saved list.

We have written a [pull request](#) to the plugin author.

QRCode

We were asked to create a new panel, from scratch, that displays a QR code linking to the current dashboard. First of all we had to get familiar with how to build a panel. To do so, we followed the instructions of one of Grafana tutorials on this website:

<https://grafana.com/tutorials/build-a-panel-plugin/#1>

As described in the tutorial, we created a plugin from a template using `grafana-toolkit`. We encountered several difficulties as errors were recently (4 days ago!) committed to the panel template we got from the tutorial. Then, with the tutorial, we understood the structure of a panel from the developer view and we added the few lines needed to generate a QR code using the `qrcode.react` package. Finally, we clarified the code and we removed all unnecessary files.

The hardest part was to create the panel, generating the QR code was done with only a few lines of code. To write down the setup instructions we tried to get the panel from the GitHub repository as a user will do. To share our plugin with the Grafana community we followed the instructions on how to Publish a plugin and made a pull request, but it didn't pass the check. We tried four different pull request but none of them pass the test and the repository was getting really messy so we stopped there for the moment. The pull request link can be found on the README.

Technologies

Grafana

Grafana is a visualization and analytics software that allows to display metrics through lots of different visualizations, each one on a Panel that can be organized on Dashboards. It's an open source software for anyone to contribute by implementing plugins, improve the documentation or develop new features on the software itself.

InfluxDB

InfluxDB is an open source time series database that handle monitoring, storing and querying datas. It helped us to get started with Grafana and to try TrackMap, the plugin we had to improve. We didn't use it after that, as we didn't need time series data for the features we have made.

JavaScript

JavaScript is an high level programming language designed to make web development. It is used with HTML and CSS to develop web front-end. Our project depends on this language as AngularJS and React are based on it.

AngularJS

AngularJS is a JavaScript-based framework for dynamic pages. Grafana was at first implemented in AngularJS. As the panel we had to improve, TrackMap, was in AngularJS we had to get familiar with it.

React

React is a JavaScript library for building user interface. In 2018 Grafana migrated from AngularJS to React for various reasons. This is why new plugins are now supposed to be developed in React. In order to develop the QR code panel we had to learn and understand basic React.

Conclusion

To go further

We would have liked to add more features in order to improve the user experience, here is a list below and why each one matters.

Trackmap

- **Change the name of an overlay.** Currently each overlay has a name which corresponds to its Leaflet id. But we think it would be convenient to be able to have a custom name. Another idea on top of the previous one could be to give the overlays from an imported file the name of the file by default instead of the Leaflet id.
- **Cancel the last overlay deleted.** There is currently no validation step before an overlay is deleted, which makes errors easy.
- **Edit an overlay.** Either by clicking on a button in the list or by clicking the overlay directly on the map, and the text would be in the text area. It could also be useful to quickly see what's in the overlay. This could seem like a big problem but a workaround is to download then delete the overlay, edit it using a text editor and upload the new version.
- **Change the order of the overlays.** The order could be changed in the list and the list order would impact the map order: the first overlay in the list would be above all the other overlays on the map. It could be particularly useful if an overlay completely covers an other layer (for example a region and a point), this way we could click on both of them.

QRCode

- **Scale the QR code to fit the panel.** This way, the user could choose the space that it wants the panel to take on the dashboard.
- **Share your dashboard via a QR code.** It would be interesting to add a button to download the QR code in the panel options in order to share it.

Get perspective

We learned a lot through this project and enjoyed working together. We used different technologies, some of them for the first time, it was a good practice because we will probably use a lot of different technologies in the future.

As it is an open source project we learned how to contribute and properly use Git but we also had to stick with the choices of the original author, for example using AngularJS (which is now old and React is preferred), and we had to understand how the original code was structured.

It was satisfying to have visual results, to create a whole new plugin and to contribute to an existing plugin with a pull request, it "feels real". It was also rewarding to be able to choose what to implement next, that motivated us.