GrafanaTrackMap

June 19, 2021

1 Using TrackMap on Grafana

We expect that one has Grafana / IndluxDB / nginx enabled on a Docker environment as explained in the manual, chapter InfluxDB.

1.1 TrackMap installation

Plese refer first to plugin's pages.

The installations is done from command line with Grafana CLI in our standalone (local) system. On Windows, the easiest way to get it with with Docker Dashboard, navigate to the `dasht_grafana` container and click on the `CLI` button. Give command:

```
/usr/share/grafana $ grafana-cli plugins install pr0ps-trackmap-panel
installing pr0ps-trackmap-panel @ 2.1.2
from: https://grafana.com/api/plugins/pr0ps-trackmap-panel/versions/2.1.2/download
into: /var/lib/grafana/plugins

  Installed pr0ps-trackmap-panel successfully

Restart grafana after installing plugins . <service grafana-server restart>
```

As suggested, close the CLI windows, head back to the Docker Dashboard and restart the container `dasht_grafana`.

Using the Grafana administrator interface, you can now observe a new, signed plugin, `TrackMap`. When you create a panel, it shows up as an alternative `Visualization` method under the corresponding drop down menu. If it does not, you may need to restart the Grafana server, which can be used simply by first stopping and the starting the entire *DashT* Docker stack.

1.2 TrackMap and Flux

Apparently, TrackMap is a GIS application and only SQL is used. However, from the Grafana's point of view it is about the query and the fields it returns.b

Somebody using TrackMap has had the issue already this year and come with a solution where the Flux itself has been used to map the fields the query produces. He came up with a solution like this:

```
from(bucket: "home_assistant")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "device_tracker.mate10")
```

```
  |> filter(fn: (r) => r["_field"] == "gps_accuracy" or r["_field"] == "latitude" or r["_field"

|> pivot(columnKey: ["_field"], rowKey: ["_time"], valueColumn: "_value")

|> duplicate(column: "latitude", as: "lat")
|> duplicate(column: "longitude", as: "lon")
|> duplicate(column: "_time", as: "tooltip")
|> duplicate(column: "velocity", as: "popup")

|> map(fn: (r) => ({ r with popup: "Speed: " + string(v:r.popup) + " km/h" }))

|> keep(columns: ["_time", "tooltip", "lat", "lon", "popup"])
```

The above creates some errors in the InfluxDB query but provides `lat` and `lon` arrays alright in *DashT* use case with Grafana.

I rather make queries for all parameters in one single panel, where I can observe the field results, say in a Table visualization. Then, in the TrackMap panel I would read from this panel and use Flux Transformation functions I can select which queries are actually executed and give new names to the fields, etc.

Here is the Grafana 7.x JSON-model of the example Dashboard used in making the TrackMap example in the manual, it looked like this in it total size. Note that only one panel is making an inquiry, others are using and transforming that inquiry:

(zoom)