Charts in the DOMIQ

In this tutorial we will present integration of the DOMIQ with the Prometheus and Grafana software. Thanks to this integration, our system has gained a whole new level of functionality to create charts.

Presenting data on charts is one of the key features of smart home. From now on you can easily create, for example, a chart of energy, water or gas consumption, view the temperature or parameters of the ventilation system in any time range. The possibilities are almost endless - you can create a chart for any variable that is present in the DOMIQ system.

1. Getting Started

This chapter describes the required devices and what role they play in order to use the new charts mechanism.

- **DOMIQ/Base module** - it will be used to define so-called metrics. Metrics are definitions of variables in the DOMIQ system that will presented on charts. If the concept of metrics is somehow fuzzy to you right now, don't worry. We'll explain it in more details later in the tutorial.
- **PC** (can be used any oneboard PC such Raspberry PI or other as well) or **external hosting server**. This device is used to install the Prometheus and Grafana software. At this stage, we would like to specify the task of both applications.
  - **Prometheus** - its main task is to collect data. Prometheus periodically polls indicated device/devices for data according to defined metrics and then collects the data in database. The polling interval is determined in the Prometheus configuration, however in Base we have put a restriction, so that polling cannot be made more often than every 15 seconds. The Base module responds to requests by returning the current state of identifiers assigned to metrics. The Prometheus you can plot charts, however its plotting functionality is very limited. That's why we use Grafana.
  - **Grafana** - is a perfect complement to the functionality of the Prometheus. It allows you to define charts of many types (eg line charts, bar charts, pie charts, etc.), while providing great editing capabilities of the presented data, including the application of mathematical formulas to these data. From Grafana's point of view, the Prometheus is a data source for charts.

  Both applications must have access to each other, so the best option is to install them on the same device. The device needs disk space for collecting data that will presented on charts. If the device will be installed in the same location as the Base module, then it should be connected to the same local network.

  In case you would like to install the Prometheus and Grafana on a external server, then you have to redirect Base's web port to the Internet to make possible for Prometheus to query for metrics data. In order to protect the Base's configuration interface we added security feature that allows Base to receive metrics queries without setting the administation access to the entire Internet (see the **Users** tab). This feature completely

blocks access to the configuration interface from outside the local network even when the web port is redirected to the Internet.

In case device works in local network, it's crucial to assign fixed IP address to it. **You also need to configure the device to run the Prometheus and Grafana at system startup.**

- **Device running the Remote app** - at the moment of writing this tutorial (February 2019), only the Remote for iOS supports the new charts mechanism. However in the nearest future, the Remote for Android will also gain this functionality. The Remote application displays charts by connecting directly to device which runs the Grafana software. If you want to preview charts from outside of local network, you need to redirect access to the device with the Grafana to the Internet (not applicable when installing on a hosting server).

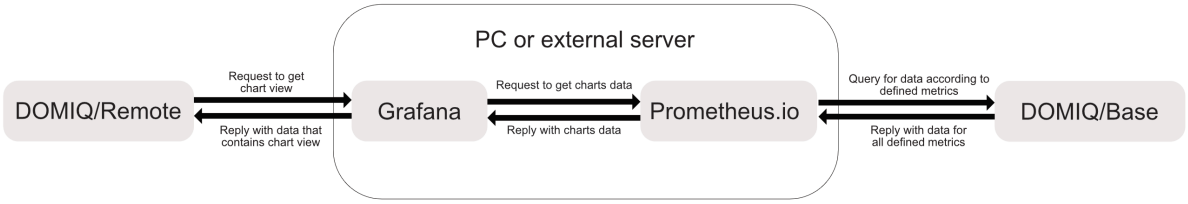


Chart preview is initiated by the Remote app, which sends a request to the Grafana software. The Grafana gets from the Prometheus all the data necessary to present chart. Received data is processed and then returned by the Grafana to the Remote as a complete chart view. Chart view is processed by the Remote and presented to user. Data exchange between the Prometheus and the Base is independent, initiated by the Prometheus and runs in defined intervals.

2. Configuration

In this tutorial we won't present installation of the Prometheus and Grafana, because this is already described in details by the software publishers. But we will explain some configuration details of both applications to make them work with the Base and Remote.

2.1. Prometheus

After the Prometheus is installated, navigate to installation directory. You'll find there the `prometheus.yml` file. Edit this file and modify configuration according to our recommendations. Below we present the `.yml` file from our test installation.

```
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any time-series scraped from this config.
  - job_name: 'base'
```

```
    # metrics_path defaults to '/metrics'
    metrics_path: '/prometheus/metrics'
    # scheme defaults to 'http'.

    static_configs:
    - targets: ['192.168.1.30:80']
```

`job_name` is the name that is used as a signature on charts legend. The name can be any without space characters, in our case we used: *base*.

`metrics_path: '/prometheus/metrics'` - the path of a URL to fetch metrics from the Base module. **This parameter must have the value as presented in the example. Otherwise, it will not be possible to get data from the Base module**.

```
 static_configs:
    - targets: ['192.168.1.30:80']
```

The address and web port number of the Base module. In case of installation on external server, here you need to enter Base's public address.

2.2. Grafana

After the Grafana is installed, navigate to installation directory. In the **Conf** folder you'll find the `custom.ini` file (this might alter depending on operating system you use). Edit this file and modify the configuration according to our recommendations.

```
# The http port to use
http_port = 3000
# The public facing domain name used to access grafana from a browser
domain =

# Default role new users will be automatically assigned (if auto_assign_org above is set to true)
auto_assign_org_role = Editor
#################################### Anonymous Auth ######################
[auth.anonymous]
# enable anonymous access
enabled = true
#################################### Basic Auth ##########################
[auth.basic]
enabled = false
```

`http_port` – here you can enter the port number on which Grafana will be reachable (you can also leave the default value).

`domain` – Leave this value empty (delete *localhost*).

`auto_assign_org_role` - change to *Editor*.

```
# enable anonymous access
enabled = true
```

If you set the `enabled` parameter to `false`, then each request sent to Grafana will require authentication (passing login and password).

```
[auth.basic]
enabled = false
```

Set to `false` to disable authentication.

2.3. Configuration of the Base

In the introduction of this tutorial, we mentioned about metrics. Metrics are mandatory so that Grafana software can display charts. Metrics are nothing else but definitions of variables that Grafana reads from the Base module. To define metrics you'll use the the **Charts** tab of the Base's configurator.

The **Charts** tab is divided into three columns: **Identifier**, **Name** and **Description**. In order to add a metric, click the **Add** button.

- In the **Identifier** column enter a variable identifier, that is available in the DOMIQ system.
- In the **Name** column you can enter a name that describes the identifier. This is optional step but we strongly recommend to do so.
  Example: suppose that the identifier *LCN.value.0.10.r1* was used. This name does not say much about the R1 variable of a LCN module with address 10. However, if you enter *temperature_bedroom* in the **Name** column, it's much easier to understand the purpose of the metric. This is a great help when defining charts in the Grafana software. If you leave the **Name** column empty, then when defining a chart in Grafana you need to refer directly to identifier name, but you should use lowercase characters and underscore characters instead of dots. **Do not enter any space characters here!**
  Example: suppose you use the *LCN.value.0.10.r1* identifier, then you should enter *lcn_value_0_10_r1* in the chart definition. Grafana software makes work easier because it has autocompletion mechanism.
- In the **Description** field you can add optional metric description.

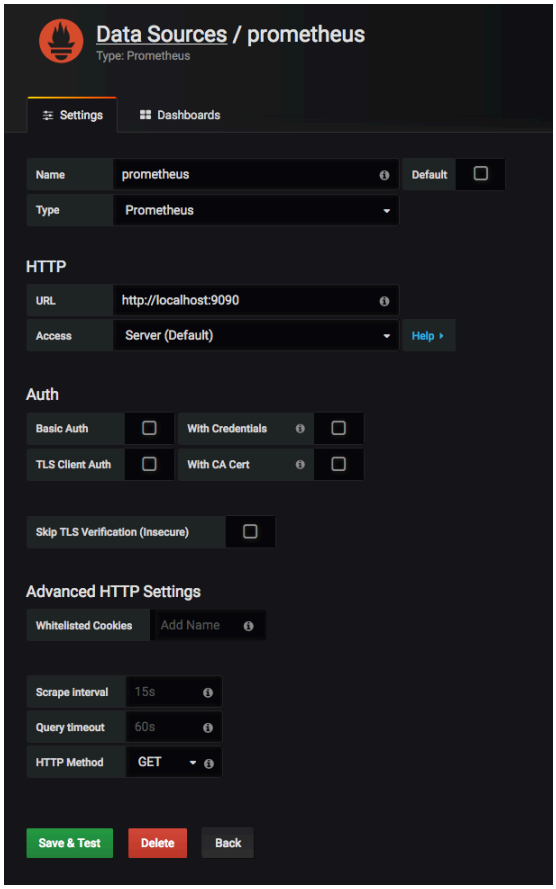
Define as many metrics as you need according to the presented procedure and then save the **Charts** tab.

3. Running Grafana

After installing and running the Prometheus and Grafana server (see the documentation of both programs), it is time to create the first chart. By default the Grafana server runs on port 3000. So in order to run Grafana configuration interface, in the web browser enter: *http://<ip_address>:3000*, where *<ip_address>* is the IP address which runs Grafana, for example: *192.168.1.100:3000*. Further in this tutorial we assume that Prometheus and Grafana both run on the same device.

3.1. Data Source Configuration

The first step is to define a data source that Grafana will use. In order to do that click: **Configuration** -> **Data Sources** -> **Add data source**. Next, configure data source as presented below:



In the **Name** field enter any name for created data source.

In the **URL** enter the address of the Prometheus server. Leave as is if you haven't made any changes to network configuration in the Prometheus.

The **Scrape interval** determines how ofter Grafana polls data from Prometheus. Do not to enter value less than 15 seconds, because the Base module doesn't allow for more frequent polling.

3.2. Dashboard Configuration

The next step is to define a dashboard which will be used to present charts. Dashboards helps to organize charts and groups them. You can have a separate dashboard for each chart type, e.g. one dashboard for all temperature charts, another for media consumption charts, etc.

To add a new dashboard click the **+** -> **Dashboard** from the sidebar. Next, click the gear icon on the topbar in order to edit dashboard settings. In the **General** section, in the **Name** field enter dashboard name.

3.3. The First Chart Panel

In order to add a new charts panel, click the chart icon on the topbar. In the main window, choose the **Graph** option. As a result you should see a new charts panel. At this moment, it shows virtual data. To fix this, click on the chart's title bar (the **Panel Title** by default) and choose the **Edit** option. Editing tools will appear below the chart.

Go to the **Metrics** tab. In the **Data Source** field, select the name of data source you created, in our case *prometheus*. The bottom part of the **Metrics** tab is used for adding metrics to a chart. Click in the text field next to the **A** letter and start typing name of any metric you defined in the Base's configurator (Grafana has autocomplete mechanism). In our case, *temperature_bedroom*. After this, click anywhere outside the textfield to confirm entered metric name. **At this point you should see your first Grafana chart**.

You can display multiple charts in a single charts panel. In order to do that you need to add more metrics to a chart. Click the **Add Query** button to add another metric. Each new metric gets next letter of the alphabet.

In the **General** tab, in the **Title** field you can add a chart title. The other tabs are for formatting (colors, scale, type of data presentation and so on). Please refer to Grafana docs and experiment to get familiar with Grafana features.

For panels that consists of several charts, it's worth defining a legend. By default, legend will be displayed at the bottom of the window. Clicking on any element of a legend will display only the chart, which this element of a legend refers to. Clicking again on the same legend item restores the visibility of all charts within this panel.

To save charts panel, click the floppy disk icon on the topbar.

LCN Temperature Chart

Temperatures in the LCN system are encoded using the following formula: `tlcn=tc*10+1000`, where `tc` is temperature in the Celsius scale and `tlcn` is temperature in the LCN scale.

The field for entering names of metrics in the **Metrics** tab (charts panel edit view), besides its basic role, has much more features that you will certainly discover when using the Grafana. One of them is the possibility to apply mathematical formulas to convert data fetched from a metric. Using this feature, we can easily convert the LCN temperature scale to the Celsius scale. Instead using just a metric name, put the following formula: (*temperature_bedroom-1000)/10*. Replace the metric name with your own.

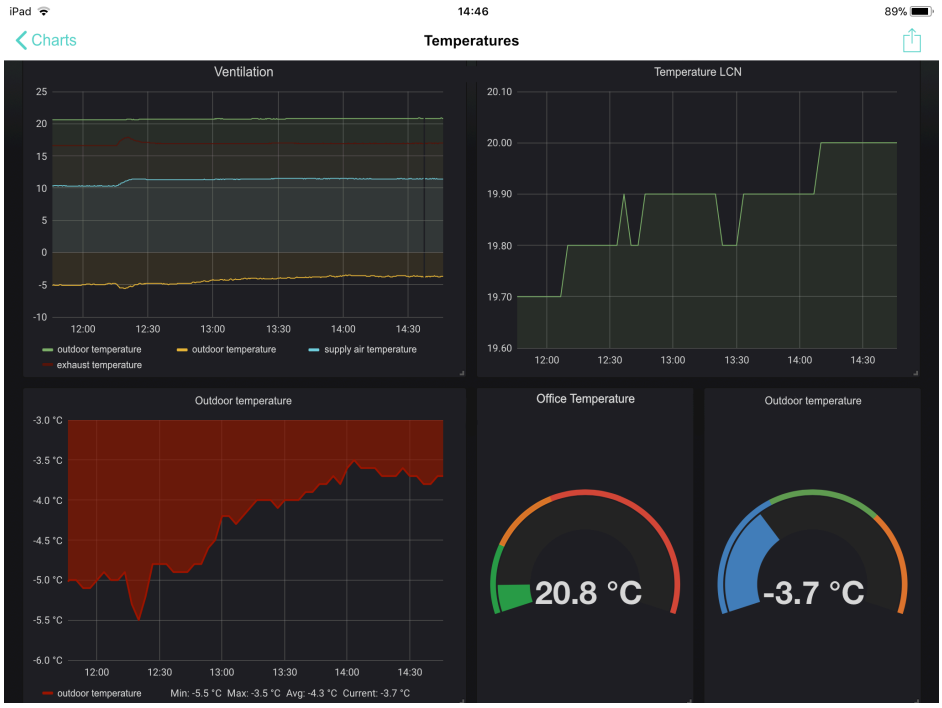
3.4. Charts in the Remote

The Remote application allows to display single charts panels (including panels containing multiple charts) and dashboards consisting of many panels as well.

Displaying a Dashboard

In the Grafana interface select the dashboard you want to display and then copy its URL from web browser. Next, go to the Base's configurator, select the **Remote** tab and add a **Chart** control. Double click on it to open edit view. In the **Local URL** paste the URL copied from web browser. If you want the dashboard to be visible outside of local network, then enter the URL that is reachable from the Internet in the **Remote URL** field.

An examplary dashboard displayed in the Remote app:

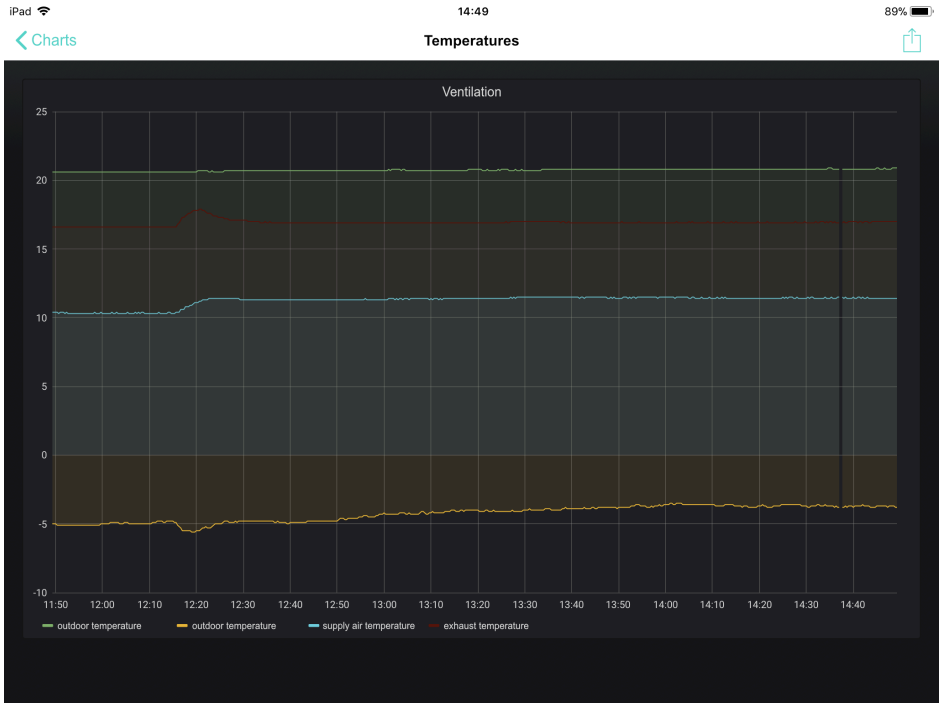


Two panels in the right bottom corner are Singlestat charts, which are available in Grafana.

Displaying a Charts Panel

In the Grafana app, go to the dashboard view, where the chart you are interested in is defined. Click on the chart's title bar and select the **View** option from the menu. Then copy the URL from web browser. The rest of the procedure is the same as for displaying a dashboard. Confirm changes by saving the **Remote** tab.

An example of a charts panel containing data from ventilation system:



Time Ranges on Charts

The **Chart** control displays chart for the last 24 hours by default. To change time range of presented data, click the icon in the right upper corner of the application and select other time range. Time range can be applied to a single charts panel or to a dashboard.

3.5. Alerts

In Grafana you can configure alerts. Alerts allow to observe and analyze data and finally to execute actions based on programmed rules. For example, an alert can be fired when observed value exceeds certain threshold. If you don't plan to analyze data in Grafana, is enough just to create events for observed variables in Base module.

It's important not to use alerts in Grafana as a replacement for events in Base!

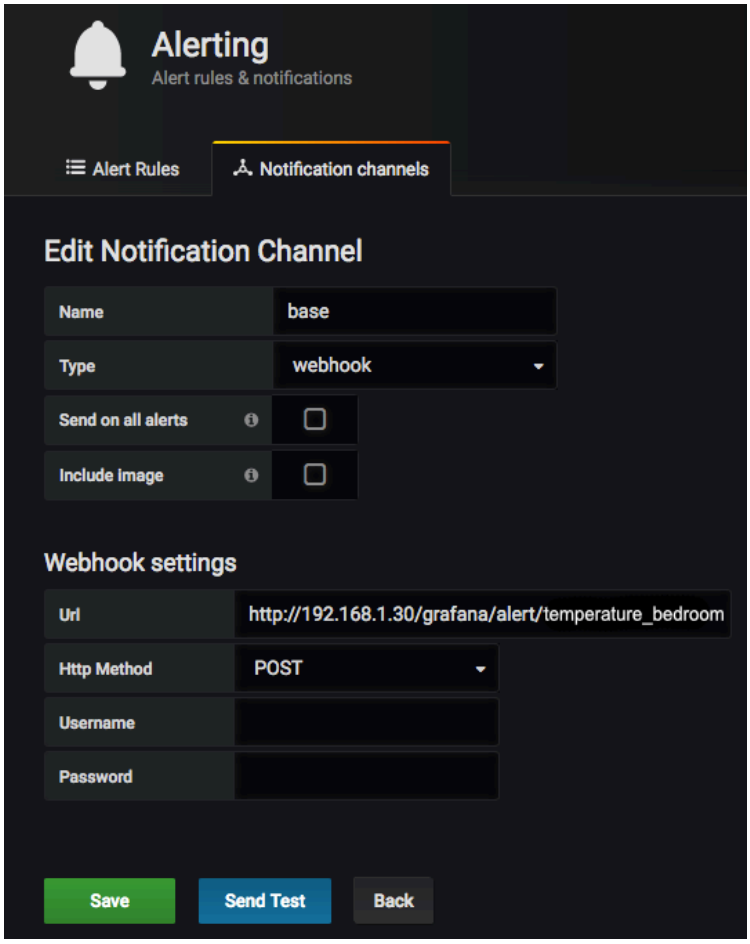
However, if the data that the Grafana reads from the Base is processed and it is necessary to inform the automation system (Base module) that an interesting event has occurred, then the alarms are what you need. There may also be a situation where data is read from another device or system, and you want to do some action in the DOMIQ system, then the alarms also become very helpful.

One of the possible actions that Grafana can do is send a HTTP request. The Base module provides a special URL address for that purpose. When Base receives a request sent to this URL it emits event, which you can use to trigger any actions.

If you do not plan to use alerts in Grafana, then our tutorial ends here. However, if you need alarms, then we invite you to read further.

3.5.1. Notification Channels

In order to send an alert from Grafana to Base it is necessary to define a notification channel. To do that, in Grafana click the bell icon on the sidebar and select the **Notification channels** option, then **New Channel**. Below is an example of a definition that sends a HTTP request to Base.



In the **Name** field enter a name for a notification channel.

In the **Type** field select **webhook**.

In the **Url** field enter the address in the following format:

http://<IP_BASE>/grafana/alert/<alert_name>, where *<IP_BASE>* should replaced with real IP address of the Base module. The *<alert_name>* can be any, but it should describe the alert.

The */grafana/alert/* part is fixed and must be included in each notification channel that suppose to be used to send alerts to Base.

The next step is to define an alarm for a specific chart. To do this, edit the charts panel that you are interested in. Then select the **Alert** tab and click **Create Alert**. In the window that will be displayed, you must define conditions for triggering the alarm (the **Alert Config** tab). In the **Notifications** tab, in the **Send to** field, select the notification channel, which the alarm should be sent to. Finally, save the chart configuration. Each time alert conditions are met it will send an alert to Base according to defined rules.

In Base this will trigger an event of the following type: *E.GRAFANA.alert = <alert_name>*, where *<alert_name>* is the name entered in notification channel definition.