



GRAFANA DOCUMENTATION

Manan Tandon

@4d616e616e

HOW TO CREATE CUSTOM PLUGIN ENVIRONMENT ?

YOU NEED TO INSTALL node.js AND yarn TO CREATE PLUGIN.

"cd" change directory to "plugins" directory inside your grafana folder.

- MAC: cd grafana/data/plugins
- Windows: cd C:\Program Files\GrafanaLabs\grafana\data\plugins

command to create custom plugins:

"npx @grafana/create-plugin your-plugin-name"

syntax:

```
[manantandon@Manans-MacBook-Air plugins % npx @grafana/create-plugin testplugin
```

after the execution of the above command Grafana will ask you certain question regarding your plugin like:

- what's the organisation name ?
- description of your plugin ?
- what type of plugin you want to make ?
- and also about your github.

```
[?] What is the organization name of your plugin? m4d616e616e
[?] How would you describe your plugin?
[?] What type of plugin would you like? panel
[?] Do you want to add Github CI and Release workflows? No
[?] Do you want to add a Github workflow for automatically checking "Grafana API compatibility" on PRs? No
```

- after you provide all the details about your plugin the setup wizard will end automatically and it will create the folder for you plugin (plugin environment) . it will also provide you with some commands to proceed further.

```
Run the following commands to get started:
```

```
* cd ./m4d616e616e-testplugin-panel
* yarn install to install frontend dependencies.
* yarn dev to build (and watch) the plugin frontend code.
```

- "cd" change directory to the created folder. example : "cd ./m4d616e616e-testplugin-panel"
- execute "yarn install" to install dependencies.
- after all the dependencies are installed execute "yarn dev"

- "cd" change directory to "**conf**" folder and edit the "**app_mode**" variable to **development** in "**defaults.ini**" file and save the file.

```
1 ##### Grafana Configuration Defaults #####
2 #
3 # Do not modify this file in grafana installs
4 #
5
6 # possible values : production, development
7 ; app_mode = production
8 app_mode = development
```

Start the **grafana-server**

MAC:

- "cd" change directory to grafana folder
- execute **./bin/grafana-server web**

syntax:

```
[manantandon@Manans-MacBook-Air grafana-9.3.6 % ./bin/grafana-server web]
```

WINDOWS:

- you can start the grafana-server from **services application**

Services (Local)					
Grafana	Name	Description	Status	Startup Type	Log On As
Start the service	Grafana	Grafana by Grafana Labs	Automatic	Automatic	Local System
	GraphicsPerfSvc	Graphics performance monitor servi...	Manual (Trigger Start)	Manual (Trigger Start)	Local System
	Group Policy Client	The service is responsible for applyi...	Automatic (Trigger Start)	Automatic (Trigger Start)	Local System
Description: Grafana by Grafana Labs	Host Network Service	Provides support for Windows Virtu...	Running	Manual (Trigger Start)	Local System
	Human Interface Device Serv...	Activates and maintains the use of ...	Running	Manual (Trigger Start)	Local System

or

- "cd" change directory to "**bin folder**" inside "**grafana folder**".
- execute **grafana-server web**

```
C:\Program Files\GrafanaLabs\grafana\bin>grafana-server web
```

go to <http://localhost:3000/> and find your **custom panel plugin** under **visualizations panel**.

The screenshot shows the Grafana dashboard editor interface. On the left, there's a large blue circle placeholder for a panel titled "Panel Title". Below it, a text input field contains the placeholder "Text option value: Default value of text input option". At the bottom, a query editor is set up with a MySQL data source, showing a single query named "A" which is a simple SELECT statement. The "Visualizations" panel on the right lists various visualization types, including "Geomap panel", "Logs", "Animated Name", and "Testplugin". The "Testplugin" item is highlighted with a red box and a red arrow points from it to a callout bubble. The callout bubble contains the text "your custom panel plugin".

your custom panel
plugin