

Plugin development

This document is not meant as a complete guide for developing plugins but more as a changelog for changes in Grafana that can impact plugin development. Whenever you as a plugin author encounter an issue with your plugin after upgrading Grafana please check here before creating an issue.

Plugin development resources

- [Grafana plugin developer guide](#)
- [Webpack Grafana plugin template project](#)
- [Simple JSON datasource plugin](#)

Changes in Grafana v4.6

This version of Grafana has big changes that will impact a limited set of plugins. We moved from systemjs to webpack for built-in plugins and everything internal. External plugins still use systemjs but now with a limited set of Grafana components they can import. Plugins can depend on libs like lodash & moment and internal components like before using the same import paths. However since everything in Grafana is no longer accessible, a few plugins could encounter issues when importing a Grafana dependency.

[List of exposed components plugins can import/require](#)

If you think we missed exposing a crucial lib or Grafana component let us know by opening an issue.

Deprecated components

The angular directive `<spectrum-picker>` is now deprecated (will still work for a version more) but we recommend plugin authors upgrade to new `<color-picker color="ctrl.color" onChange="ctrl.onSparklineColorChange"></color-picker>`

Changes in Grafana v6.0

DashboardSrv.ts

If you utilize [DashboardSrv](#) in your plugin code, `dash` was renamed to `dashboard`.