

# Package hierarchy

The Go package hierarchy in Grafana should be organized logically (Ben Johnson's [article](#) served as inspiration), according to the following principles:

- Domain types and interfaces should be in "root" packages (not necessarily at the very top, of the hierarchy, but logical roots)
- Sub-packages should depend on roots - sub-packages here typically contain implementations, for example of services

## Practical example

The `pkg/plugins` package contains plugin domain types, for example `DataPlugin`, and also interfaces such as `RequestHandler`. Then you have the `pkg/plugins/managers` subpackage, which contains concrete implementations such as the service `PluginManager`. The subpackage `pkg/plugins/backendplugin/coreplugin` contains `plugins.DataPlugin` implementations.