

Scopes, Contexts and Hooks

Scopes, Hooks, and Contexts

Scopes

Ignition defines three different **scopes**: the *gateway*, the *designer*, and the *client*. The scopes are used in various ways, such as defining which JARs should be loaded by each piece and the addressing of push messages. A module may include pieces for multiple scopes, and resources in a module can be assigned to multiple scopes at once.

For each scope that the module wishes to run in, it will define a **module hook**. This is the entry point for the module in that scope, and provides several lifecycle functions. Each scope provides a **context** to the hook, which can be used to access all of the platform services of that scope.

Module Hooks

There are three interfaces that define the hooks, that correspond to the scopes:

- GatewayModuleHook
- DesignerModuleHook
- ClientModuleHook

In practice, it is best to extend from the Abstract implementations of these hooks (for example, `AbstractGatewayModuleHook`) instead of implementing the interfaces yourself. Each module hook must be defined in the module descriptor file, as created by the IgnitionSDK Maven Plugin or created in accordance with the [Module Descriptor Specification](#).

Contexts

The *context* for the given scope will be passed to the setup or startup method of the associated hook. The modules should hold on to these contexts and pass them to subsystems that they define as it is the primary way to access the services provided by the Ignition platform.