Sommario

1. Inviare richieste via telegram

vedere questo sito per info e esempi

https://www.raspberryitaly.com/rpinotify/

2. Comprendere la struttura del file [Unit] (.service)

Tratto dal documento originale

Un file Unit è costituito tipicamente da 3 sezioni:

```
[Unit] — contains generic options that are not dependent on the type of the unit.
These options provide unit description, specify the unit's behavior, and set
dependencies to other units. For a list of most frequently used [Unit] options, see
Table 8.9, “Important [Unit] Section Options”.\\
[unit type] — if a unit has type-specific directives, these are grouped under a
section named after the unit type. For example, service unit files contain the
[Service] section, see Table 8.10, “Important [Service] Section Options” for most
frequently used [Service] options.\\
[Install] — contains information about unit installation used by systemctl enable
and disable commands, see Table 8.11, “Important [Install] Section Options” for a
list of [Install] options.
```

Table 8.9. Important [Unit] Section Options
Option [a] Description
Description A meaningful description of the unit. This text is displayed for example in the output of the systemctl status command.
Documentation Provides a list of URIs referencing documentation for the unit.
After [b] Defines the order in which units are started. The unit starts only after the units specified in After are active. Unlike Requires, After does not explicitly activate the specified units. The Before option has the opposite functionality to After.
Requires Configures dependencies on other units. The units listed in Requires are activated together with the unit. If any of the required units fail to start, the unit is not activated.
Wants Configures weaker dependencies than Requires. If any of the listed units does not start successfully, it has no impact on the unit activation. This is the recommended way to establish custom unit dependencies.
Conflicts Configures negative dependencies, an opposite to Requires.
[a] For a complete list of options configurable in the [Unit] section, see the systemd.unit(5) manual page.
[b] In most cases, it is sufficient to specify just the ordering dependencies with After and Before unit file options. If you decide to set a requirement dependency with Wants (recommended) or Requires, usually it must be supplemented with After as ordering and requirement dependencies are independent from each other. Note that for sake of simplicity and flexibility, it is recommended to use the dependency options reasonably and instead rely on techniques such as bus-based or socket-based activation when possible.

Table 8.10. Important [Service] Section Options
Option [a] Description
Type Configures the unit process startup type that affects the functionality of ExecStart and related options.

One of:

```
simple – The default value. The process started with ExecStart is the main process
of the service.\\
forking – The process started with ExecStart spawns a child process that becomes
the main process of the service. The parent process exits when the startup is
complete.\\
oneshot – This type is similar to simple, but the process exits before starting
consequent units.\\
dbus – This type is similar to simple, but consequent units are started only after
the main process gains a D-Bus name.\\
notify – This type is similar to simple, but consequent units are started only
after a notification message is sent via the sd_notify() function.\\
idle – similar to simple, the actual execution of the service binary is delayed
until all jobs are finished, which avoids mixing the status output with shell
output of services.
```

ExecStart Specifies commands or scripts to be executed when the unit is started. ExecStartPre and ExecStartPost specify custom commands to be executed before and after ExecStart. Type=oneshot enables specifying multiple custom commands that are then executed sequentially.
ExecStop Specifies commands or scripts to be executed when the unit is stopped.
ExecReload Specifies commands or scripts to be executed when the unit is reloaded.
Restart With this option enabled, the service is restarted after its process exits, with the exception of a clean stop by the systemctl command.
RemainAfterExit If set to True, the service is considered active even when all its processes exited. Default value is False. This option is especially useful if Type=oneshot is configured.
[a] For a complete list of options configurable in the [Service] section, see the systemd.service(5) manual page.

Table 8.11. Important [Install] Section Options
Option [a] Description
Alias Provides a space-separated list of additional names for the unit.
RequiredBy A list of units that depend on the unit. When this unit is enabled, the units listed in RequiredBy gain a Require dependency on the unit.
WantedBy A list of units that weakly depend on the unit. When this unit is enabled, the units listed in WantedBy gain a Want dependency on the unit.
Also Specifies a list of units to be installed or uninstalled along with the unit.
DefaultInstance Limited to instantiated units, this option specifies the default instance for which the unit is enabled. See Section 8.6.5, “Working with Instantiated Units”
[a] For a complete list of options configurable in the [Install] section, see the systemd.unit(5) manual page.
A whole range of options that can be used to fine tune the unit configuration, Example 8.17, “postfix.service Unit File” shows an example of a service unit installed on the system. Moreover, unit file options can be defined in a way that enables dynamic creation of units as described in Section 8.6.5, “Working with Instantiated Units”.