

Logging

Description

The `npm` CLI has various mechanisms for showing different levels of information back to end-users for certain commands, configurations & environments.

Setting Log Levels

loglevel `loglevel` is a global argument/config that can be set to determine the type of information to be displayed.

The default value of `loglevel` is "notice" but there are several levels/types of logs available, including:

- "silent"
- "error"
- "warn"
- "notice"
- "http"
- "timing"
- "info"
- "verbose"
- "silly"

All logs pertaining to a level proceeding the current setting will be shown.

All logs are written to a debug log, with the path to that file printed if the execution of a command fails.

Aliases The log levels listed above have various corresponding aliases, including:

- `-d: --loglevel info`
- `--dd: --loglevel verbose`
- `--verbose: --loglevel verbose`
- `--ddd: --loglevel silly`
- `-q: --loglevel warn`
- `--quiet: --loglevel warn`
- `-s: --loglevel silent`
- `--silent: --loglevel silent`

foreground-scripts The `npm` CLI began hiding the output of lifecycle scripts for `npm install` as of `v7`. Notably, this means you will not see logs/output from packages that may be using “install scripts” to display information back to you or from your own project’s scripts defined in `package.json`. If you’d like to change this behavior & log this output you can set **foreground-scripts** to `true`.

Registry Response Headers

npm-notice The `npm` CLI reads from & logs any **npm-notice** headers that are returned from the configured registry. This mechanism can be used by third-party registries to provide useful information when network-dependent requests occur.

This header is not cached, and will not be logged if the request is served from the cache.

See also

- `config`