Synopsis

```
npm ci

aliases: clean-install, ic, install-clean, isntall-clean
```

Description

This command is similar to `npm install`, except it's meant to be used in automated environments such as test platforms, continuous integration, and deployment -- or any situation where you want to make sure you're doing a clean install of your dependencies.

`npm ci` will be significantly faster when:

- There is a `package-lock.json` or `npm-shrinkwrap.json` file.
- The `node_modules` folder is missing or empty.

In short, the main differences between using `npm install` and `npm ci` are:

- The project **must** have an existing `package-lock.json` or `npm-shrinkwrap.json`.
- If dependencies in the package lock do not match those in `package.json`, `npm ci` will exit with an error, instead of updating the package lock.
- `npm ci` can only install entire projects at a time: individual dependencies cannot be added with this command.
- If a `node_modules` is already present, it will be automatically removed before `npm ci` begins its install.
- It will never write to `package.json` or any of the package-locks: installs are essentially frozen.

Example

Make sure you have a package-lock and an up-to-date install:

```
$ cd ./my/npm/project
$ npm install
added 154 packages in 10s
$ ls | grep package-lock
```

Run `npm ci` in that project

```
$ npm ci
added 154 packages in 5s
```

Configure Travis to build using `npm ci` instead of `npm install`:

```
# .travis.yml
install:
- npm ci
# keep the npm cache around to speed up installs
cache:
  directories:
  - "$HOME/.npm"
```

Configuration

`audit`

- Default: true
- Type: Boolean

When "true" submit audit reports alongside the current npm command to the default registry and all registries configured for scopes. See the documentation for `npm_audit` for details on what is submitted.

`foreground-scripts`

- Default: false
- Type: Boolean

Run all build scripts (ie, `preinstall`, `install`, and `postinstall`) scripts for installed packages in the foreground process, sharing standard input, output, and error with the main npm process.

Note that this will generally make installs run slower, and be much noisier, but can be useful for debugging.

`ignore-scripts`

- Default: false
- Type: Boolean

If true, npm does not run scripts specified in package.json files.

Note that commands explicitly intended to run a particular script, such as `npm start`, `npm stop`, `npm restart`, `npm test`, and `npm run-script` will still run their intended script if `ignore-scripts` is set, but they will *not* run any pre- or post-scripts.

`script-shell`

- Default: '/bin/sh' on POSIX systems, 'cmd.exe' on Windows
- Type: null or String

The shell to use for scripts run with the `npm exec`, `npm run` and `npm init <pkg>` commands.

See Also

- [npm install](#)
- [package-lock.json](#)