2022-03-02

NPM Tools

A short introduction to a collection of useful node packages for everyday use.

nvm

nvm is a version manager for node.js

Why?

> nvm allows you to quickly install and use different versions of node via the command line on **all major operating systems**

- POSIX-compliant shell: sh, dash, ksh, zsh, bash
- platforms: unix, macOS, and windows WSL
- simply install/uninstall node version(s)
- have multiple node versions installed
- pin node version used in project by adding a `.nvmrc`

husky

"modern native git hooks made easy" – Easily setup local git commit hooks for the whole team.

git hooks?

- Run custom script(s) every time an action occurs (f.e. "commit", "push")
- use cases:
  - lint file(s)
  - auto-format file(s)
  - run test(s)
  - build files
  - generate types
  - validate format of commit-message

Setup

Install the dependency:

```
npm install husky --save-dev;
```

Set the "prepare" script in `package.json`, install git-hooks:

```
npm set-script prepare "husky install"
npm run prepare
```

This will also create `.husky` directory and add it to `.gitignore`.

Example: pre-push

Run a script everytime new code is pushed to a remote with the `pre-push` action:

```
npx husky add .husky/pre-push "npm lint"
git add .husky/pre-push
```

```
#!/bin/sh
. "$(dirname "$0")/_/husky.sh"

# <insert lint command here>
```

bypass hooks

Sometimes scripts are slow and must be bypassed:

```
git commit --no-verify -m "chore: my message"
```

npm-check

"Check for outdated, incorrect, and unused dependencies." - Interactive way of updating npm packages

Why?

- get an overview of the up-to-date "state" of dependencies
- visually appealing, simple way of updating multiple packages
- see different types of updates: patch, minor, major
- find extranous packages
- find missing packages
-


~/projects/deskdrive ✗ master ❯ npm-check -

📦 depcheck

Analyze dependencies in a project and find unused and useless dependencies.

Use-Cases

- ideal for integration into CI
- detect packages that have been added by mistake
- detect packages which are not actually used (anymore)

npx depcheck

```
UPDATE! Your local install is out of date. https://github.com/aws/aws-sdk-js
        npm install --save aws-sdk@2.990.0 to go from 2.986.0 to 2.990.0

NEW VER! NonSemver update available. https://axios-http.com
        npm install --save axios@0.21.4 to go from 0.20.0 to 0.21.4

MAJOR UP Major update available. https://github.com/tj/commander.js#readme
        npm install --save commander@8.2.0 to go from 6.2.1 to 8.2.0

NOTUSED? Still using express-session?
        Depcheck did not find code similar to require('express-session') or import from 'exp
        Check your code before removing as depcheck isn't able to foresee all ways dependenc
        Use --skip-unused to skip this check.
        To remove this package: npm uninstall --save express-session

MAJOR UP Major update available. https://github.com/prettymuchbryce/node-http-status#readme
        npm install --save http-status-codes@2.1.4 to go from 1.4.0 to 2.1.4

MAJOR UP Major update available. https://github.com/mongodb/node-mongodb-native
        npm install --save mongodb@4.1.2 to go from 3.7.0 to 4.1.2

NOTUSED? Still using mongodb-uri?
        Depcheck did not find code similar to require('mongodb-uri') or import from 'mongodb
        Check your code before removing as depcheck isn't able to foresee all ways dependenc
        Use --skip-unused to skip this check.
        To remove this package: npm uninstall --save mongodb-uri

MAJOR UP Major update available. https://mongoosejs.com
        npm install --save mongoose@6.0.6 to go from 5.13.9 to 6.0.6

MAJOR UP Major update available. https://github.com/pusher/pusher-rest-node#readme
        npm install --save pusher@5.0.0 to go from 4.0.2 to 5.0.0

UPDATE! Your local install is out of date. https://gitlab.com/catamphetamine/read-excel-fil
        npm install --save read-excel-file@5.2.10 to go from 5.2.9 to 5.2.10

MAJOR UP Major update available. https://github.com/socketio/socket.io#readme
```

False Positive(s)

Some dependencies are not required in the sources but are used for compilation, linting and other tasks. Those packages often are detected and are false-positives and can be ignore while also documenting why they are added:

```
# .depcheck.yml
# use `npx depcheck --config .depcheck.yml`
ignores:
  # required by ngx-bootstrap
  - "bootstrap"
  # used for git hooks
  - "husky"
```

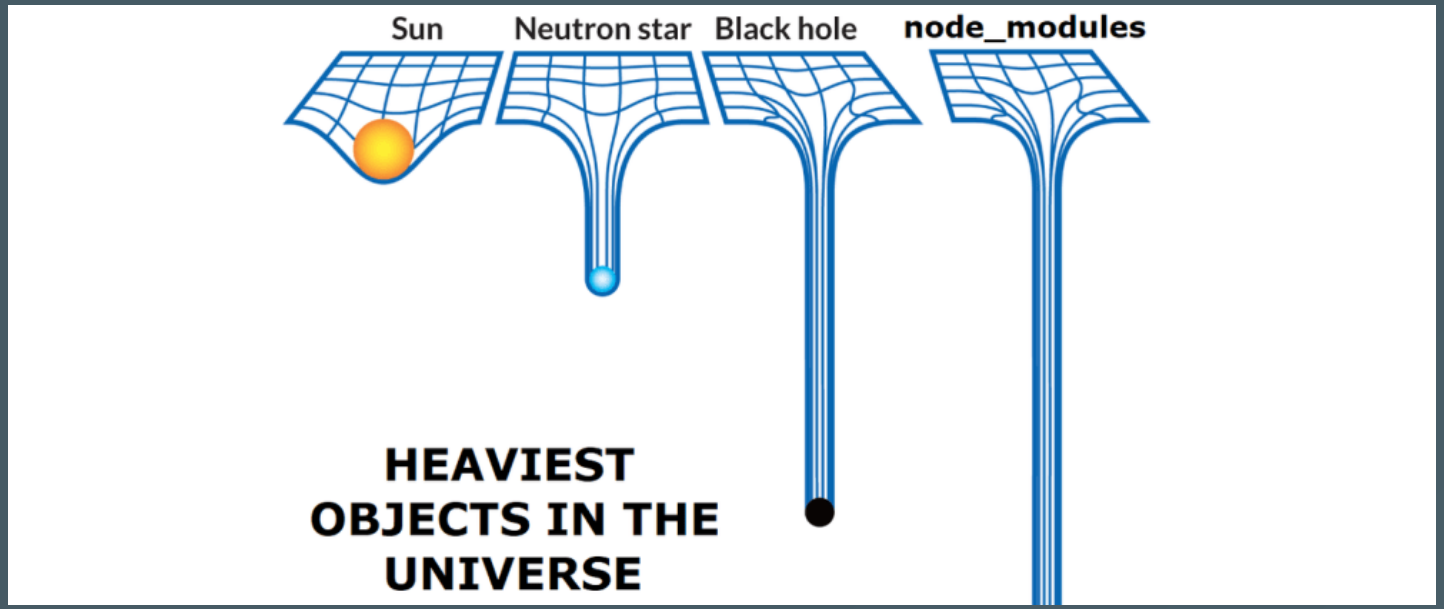
modclean

"Remove unwanted files and directories from your node_modules folder"

Why?

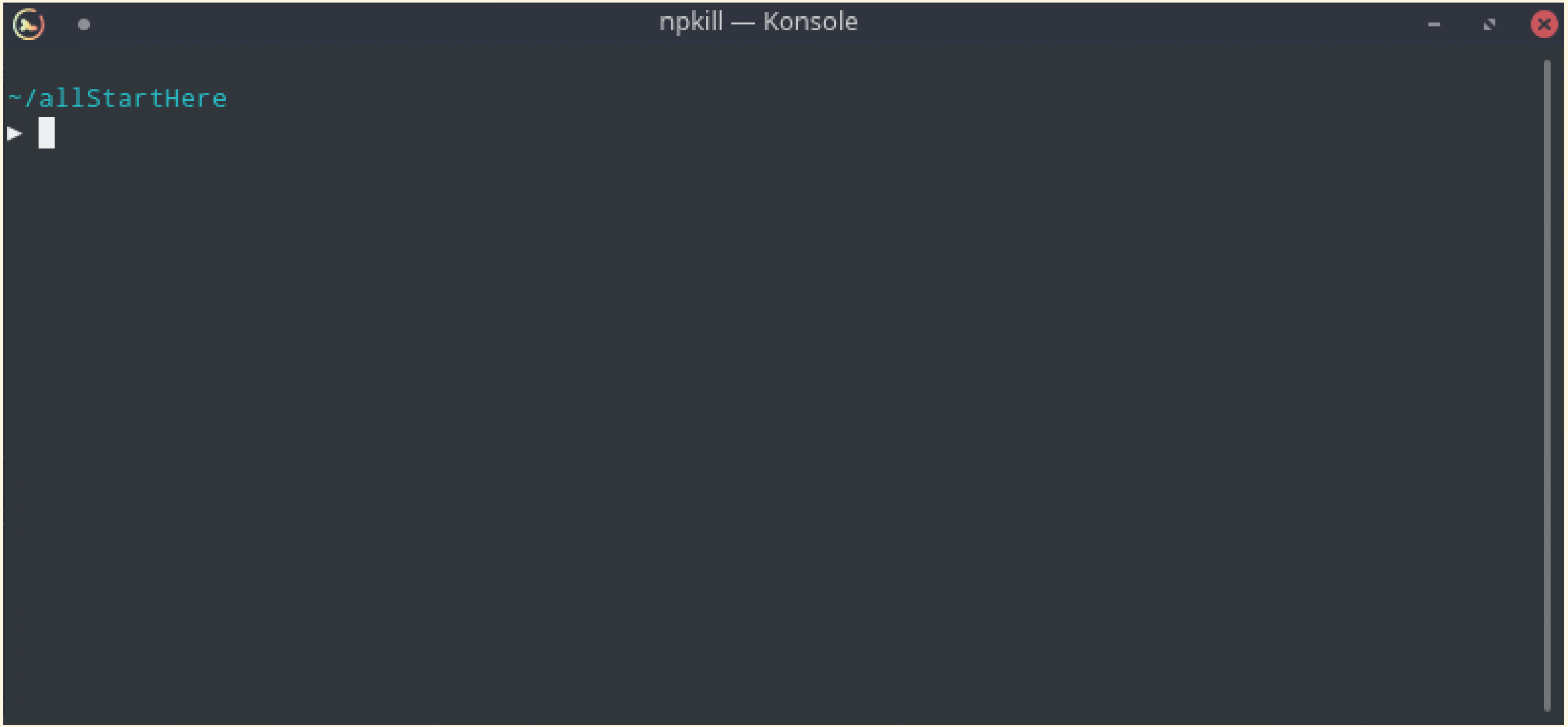
Collection of different glob-patterns & filters to remove files from node_modules which are not used.

- reduce size of packaged applications (electron)
- reduce size of docker image
- in CI reduce space used for cache(s)
- save space on machine
- Example Benchmarks



npkill

"Easily find and remove old and heavy node_modules folders ✨"


npkill — Konsole
~/allStartHere

license-checker-rseidelsohn

"Ever needed to see all the license info for a module and its dependencies?"

Why?

- extract SPDX identifiers of packages
- list licenses for legal auditing, documentation:

```
npx license-checker-rseidelsohn --csv --out /path/to/licenses.csv
```

- check if packages added with unwanted licenses

```
npx license-checker-rseidelsohn --failOn 'GPL'
```

Others?

- check packages in `package.json` files of projects that you're using to find new inspiring helpful tools.
- check github explore: https://github.com/topics/javascript
- check node weekly newsletter

Thanks for listening!