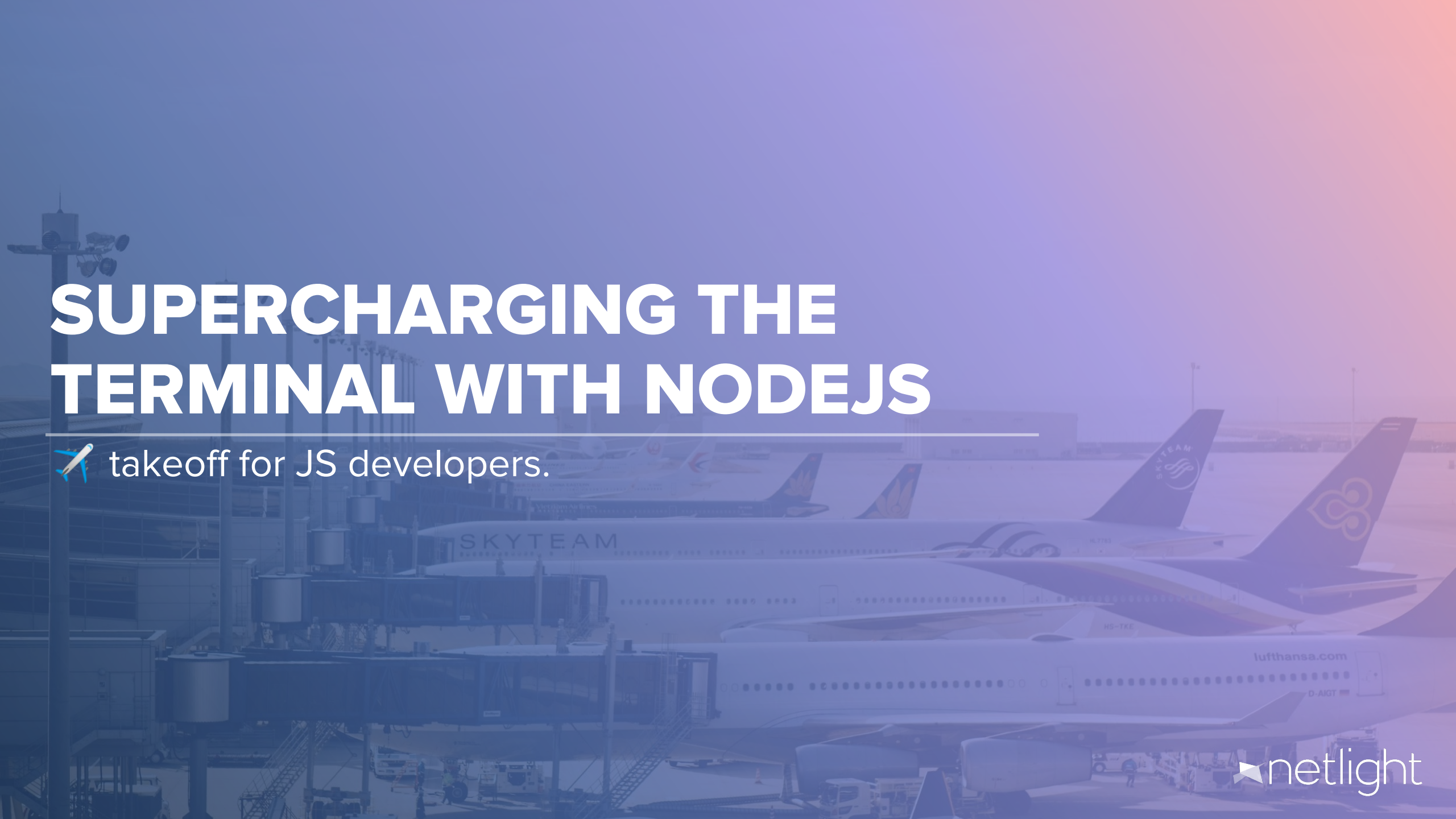

SUPERCHARGING THE
TERMINAL WITH NODEJS
takeoff for JS developers.
SKYTEAM
lufthansa.com
netlight

HENDRIK

JavaScript enthusiast

Developer for fun 🎉

netlight

```
hewa@ham-hewa MINGW64 ~/dev
$ cd http-status/

hewa@ham-hewa MINGW64 ~/dev/http-status (master)
$ ls
codes.json  index.js*  node_modules/  package.json  README.md

hewa@ham-hewa MINGW64 ~/dev/http-status (master)
$ cat package.json | pjs -f "match(/version/)" -m "split('\"')[3]"
1.0.0

hewa@ham-hewa MINGW64 ~/dev/http-status (master)
$
```

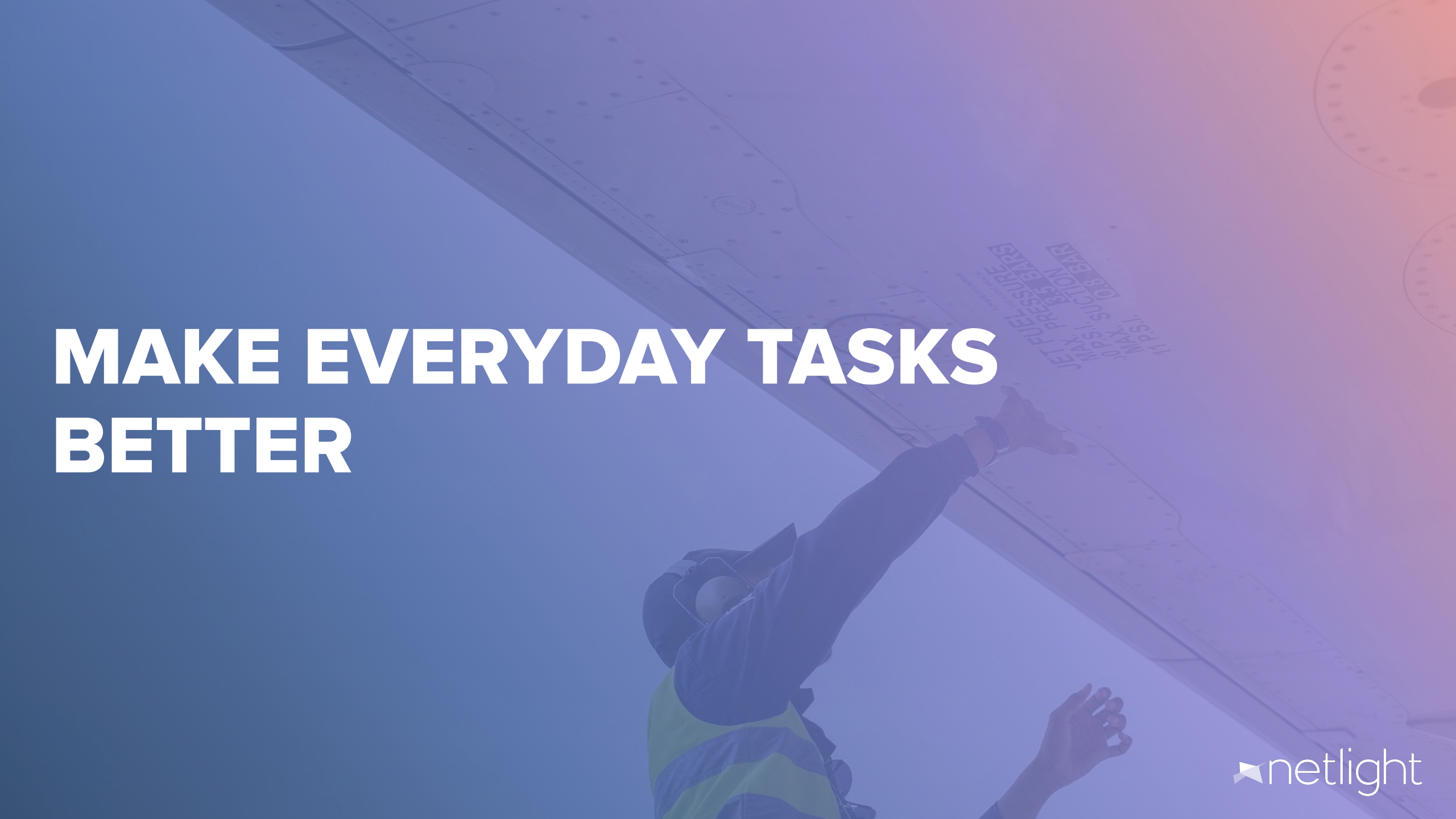
- **Terminal**

Development needs it, we love it, we hate it, we use it!

CLIs with NodeJS

- You can build a CLI in Node
- JS developers can read that

MAKE EVERYDAY TASKS BETTER



netlight

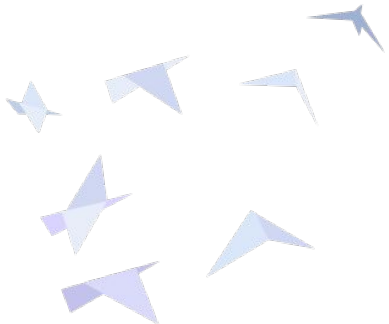
Make everyday tasks bestter

- Utility:
  - Vtop – An amazing version of top
  - Speed-test – Test you internet speed
  - Brightness-cli – Change your screen brightness
  - Alder – Tree with colors

Tree using alder

```
hewa@ham-hewa MINGW64 ~/dev/http-status (master)
$ alder

.
├── codes.json
├── index.js
├── package.json
└── README.md
1 directories, 5 files (16.7 kB)
```



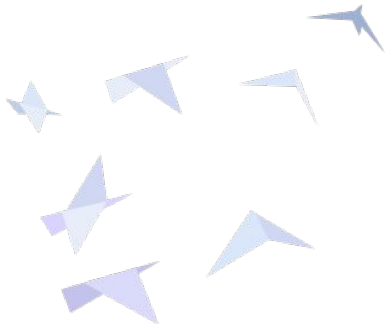
Make everyday tasks better

- Jsome – Nice JSON in your terminal
- Lessmd – Colored Markdown in the terminal
- Hget – Grab a text version of a website
- Pjs – Filter using JavaScript

Filter output with pjs

```
hewa@ham-hewa MINGW64 ~/dev/http-status (master)
$ jsome package.json
{
  name: "@hoverbaum/http-code",
  version: "1.0.0",
  description: "A super simple CLI to look up the meaning of http status codes.",
  main: "index.js",
  author: "Hendrik Wallbaum <mail@hendrikwallbaum.de>",
  dependencies: {
    commander: "^2.14.1"
  },
  bin: "index.js"
}

hewa@ham-hewa MINGW64 ~/dev/http-status (master)
$ cat package.json | pjs -f "match(/version/)" -m "split('\"')[3]"
1.0.0
```




BEYOND THE TERMINAL
netlight

Beyond the terminal

- Diff2html – See git diff in browser
- Live-server – Preview websites in your browser
- Bcat – Cat to browser
- Ttystudio – Record terminal to gif
- Pagers – Screenshots of websites
- Pen – Markdown preview in browser

DO FUN THINGS

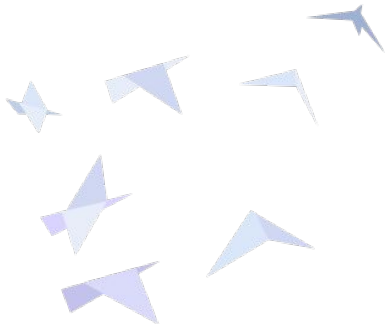
netlight

Do fun things

- Emoj – Find relevant emojies
- Normit – Translate text in your terminal
- Weather-cli – Up to date weather in on command
- Movie – Get information about a movie
- Vaca – Generate a random cow

Do fun things

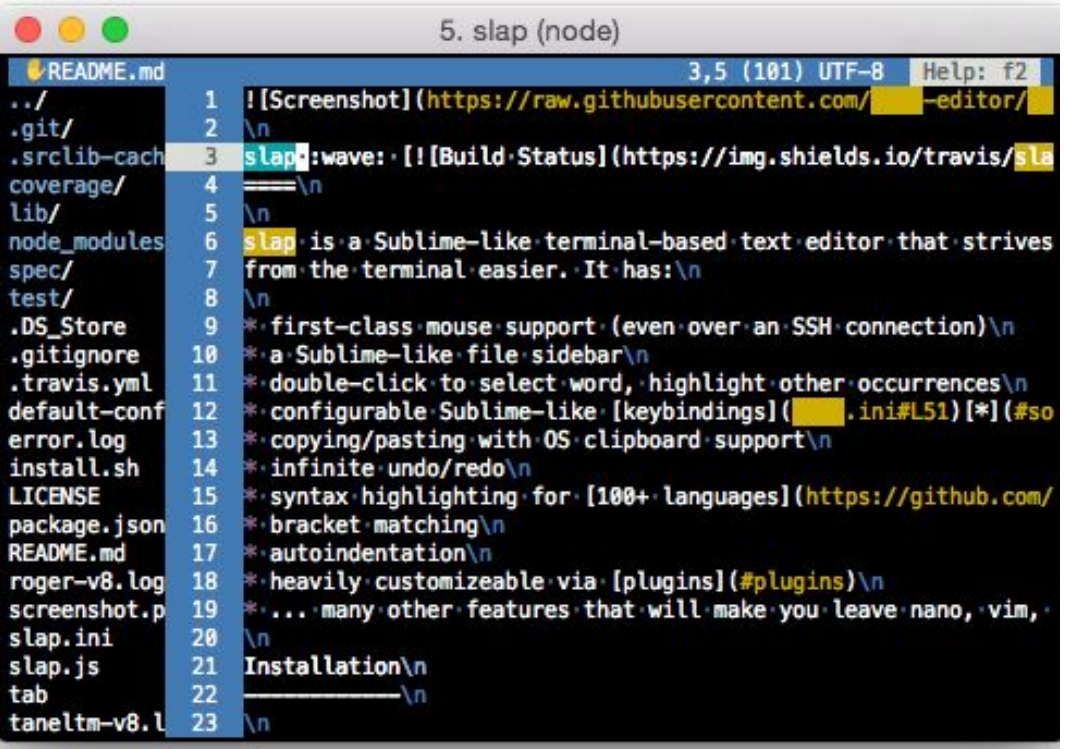
```
hewa@ham-hewa MINGW64 ~/dev/http-status (master)
$ vaca
```




AMAZING POTENTIAL
netlight

•Node based terminal

- Full featured Unix like terminal
- Compatible with all platforms
- Just use the commands



•Slap

- Node based editor in your terminal
- Mouse support
- Sidebar with file tree
- Syntax highlighting, plugins

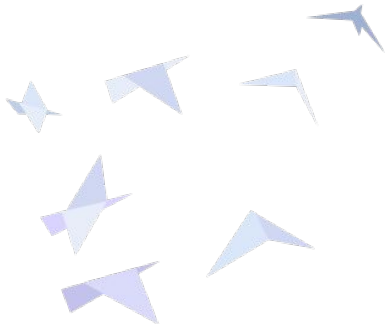
netlight

HTTP-Codes

```
hewa@ham-hewa MINGW64 ~/dev/http-status (master)
$ http-code 404

404: Not Found

The server can not find requested resource. In the browser, this means the URL is not recognized. In an API, this can also me
an that the endpoint is valid but the resource itself does not exist. Servers may also send this response instead of 403 to h
ide the existence of a resource from an unauthorized client. This response code is probably the most famous one due to its fr
equent occurence on the web.
```



```
hewa@ham-hewa MINGW64 ~/dev/http-status (master)
$ alder --exclude=commander

.
├── codes.json
├── index.js
├── node_modules
├── package.json
└── README.md
3 directories, 5 files (16.6 kB)
```

HTTP-Codes

- Display information about a given HTTP status code.

Idea:

- JSON file with information
- Handle user input and find right information to display
- Otherwise show a message

index.js

```
#! /usr/bin/env node

const cli = require('commander')
let statusCode = null

cli
  .version(require('./package.json').version)
  .arguments('<status code>')
  .action(code => {
    statusCode = code
  })
  .parse(process.argv)

if (!statusCode) {
  console.log('No status Code supplied. Please specify a status code you are looking for.')
  process.exit(1)
}

const codes = require('./codes.json')
statusCode = parseInt(statusCode)

const code = codes.find(code => code.number === statusCode)
if (!code) {
  console.log('Could not find the code you are looking for.\nIf you think it should exist please consider opening an Issue.')
  process.exit(1)
}

console.log(`
${code.number}: ${code.name}

${code.description}
`)

```

•Dive in

- Commander for CLI
- Define usage
- Parse arguments
- Check input
- Provide information

index.js

```
#! /usr/bin/env node

const cli = require('commander')
let statusCode = null

cli
  .version(require('./package.json').version)
  .arguments('<status code>')
  .action(code => {
    statusCode = code
  })
  .parse(process.argv)

if (!statusCode) {
  console.log('No status Code supplied. Please specify a status code you are looking for.')
  process.exit(1)
}

const codes = require('./codes.json')
statusCode = parseInt(statusCode)

const code = codes.find(code => code.number === statusCode)
if (!code) {
  console.log('Could not find the code you are looking for.\nIf you think it should exist please consider opening an Issue.')
  process.exit(1)
}

console.log(`
${code.number}: ${code.name}

${code.description}
`)

```

Further reading

- Lists of amazing tools:
  - ahttps://github.com/sindresorhus/awesome-nodejs#command-line-apps
  - https://github.com/agarrharr/awesome-cli-apps
- Winston – https://github.com/tj/commander.js
- Vorpal – https://github.com/dthree/vorpal
- Pictures from: https://unsplash.com/

HENDRIK WALLBAUM

hendrik.wallbaum@netlight.com
@hoverbaum

netlight.com

netlight