Command Line Tools with Node.js

About me

I'm a developer with a passion for JavaScript, Hackathons and good Whisky.

Follow me: fb.com/dominik402 | @dkundel | github/dkundel

Why command line tools?

The terminal is badass!


It gives you super powers!

Command line tools

The ultimate productivity tool

- Build on existing and built-in tools
- Piping (|)
- Redirects (> / <)
- Chaining (&&)
- Parallel execution (&)

Writing a new script

The difficult choice of a language

Why Node.JS?

- Cross-platform
- Leverage over 200k exisiting libraries
- Easy distribution thanks to NPM
- Streams and event driven by design
- Most likely already installed

`unicorn [options] <recipients...>`

- Read content piped into the tool
- Send emails with the content to various recipients


Let's
Code!

Consume your new command line tool

Install your script globally

```
> npm install unicorn --global
> unicorn --help
  Usage: example [options] <recipients...>

  Options:

    -h, --help                output usage information
    -V, --version             output the version number
    -s, --subject [name]      Subject
```

Use your tool in a project

```
› npm install typescript --save-dev
```

Creates an entry:

`node_modules/.bin/tsc`

Use your tool in a project

```
{
  ...
  "scripts": {
    "build": "tsc",
    "postinstall": "npm run build"
  }
  ...
}
```

```
> npm run build
```

Pass additional arguments

```
{
  ...
  "scripts": {
    "build": "tsc",
    "build:dev": "npm run build -- --watch",
    "postinstall": "npm run build"
  }
  ...
}
```

```
> npm run build:help
```

Result: A super easy project setup

```
> git clone https://github.com/project/example.git
> cd example && npm install
```

Thanks for listening!

Questions?

1. Create a new project

```
> mkdir unicorn && cd $_
> npm init
```

2. Change the `package.json`

```
{
  ...
  "bin": {
    "unicorn": "unicorn.js"
  },
  ...
}
```

3. Install some dependencies

```
> npm install commander nodemailer --save
```

4. Create `unicorn.js` and add dependencies

```
const pkg = require('./package.json');
const nodemailer = require('nodemailer');
let program = require('commander');
```

5. Read piped content from stdin

```
//...
let input = '';
process.stdin.resume();
process.stdin.setEncoding('utf8');
process.stdin.on('data', (data) => {
  input += data;
});

process.stdin.on('end', () => {
  //...
}
//...
```

6. Read command line arguments

```
//...
program
  .version(pkg.version)
  .option('-s, --subject [text]', 'Subject')
  .parse(process.argv);

console.log('Subject specified: %s', program.subject);
//...
```

7. Load config

```
//...
const os = require('os');
const config = require(`${os.homedir()}/unicorn.config.json`);
//..
```

8. Send email

```
//...
let transport = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: config.username,
    pass: config.password
  }
});

transport.sendMail({
  from: config.username,
  to: recipient,
  subject: subject,
  html: input
}, function (err, data) {
  // callback
});
```

9. Exit the program

```
process.exit(0); // success
process.exit(1); // failure
```

10. Color output

```
const chalk = require('chalk');
const error = chalk.bold.error;
//...
console.error('%s Some failure', error('ERROR'));
process.exit(1);
```