

NPM

*The only build too you need**

**probably*



ZAKIYA KHABIR

Chapter Three

@zakiyadesigns

CHAPTER THREE IS HIRING

WHO ARE YOU?

QUESTIONS?

INTRO



WHY TASK RUNNERS?

.....

- At Minimum
 - Sass
 - Version control
 - Watch
- Extras
 - minify files
 - concatenate files
 - autoprefixer for compatibility
 - generate svg sprite
 - unit tests
 - optimize files



WHAT DOES THIS HAVE TO DO WITH DRUPAL?

-
- Nothing!
 - Everything!
 - A note about libraries!

WHAT IS NPM?

- Node Package Manager is a package manager for NodeJs
- A set of javascript modules that can run server and client side scripts
- Depends on and comes with NodeJs
- Packages can be entire frameworks (Angular) or simple scripts

**WAIT I THOUGHT PEOPLE
WERE USING GRUNT OR
GULP OR BROCCOLI OR
COMPASS OR MIMOSA OR
CAKE OR BOWER OR
BRUNCH ...?**



WHAT ABOUT GULP?

- Fast
- Robust
- Arguably most popular today
- Let's look at pros and cons.

PROS OF NPM OVER GULP

- doesn't fail silently
- reduces complexity
- fewer layers of abstraction
- slowly developers are moving away from Gulp
- Gulp and Grunt often run NPM packages that are wrappers for other NPM projects.

CONS OF NPM OVER GULP

- harder to find script specific documentation (look for CLI)
- some (resolvable) compatibility issues with Windows
- can't comment a json file
- no variables, elaborate config
- duplicate/similar packages can be confusing
- ugly error messages

INSTALL & USE

CODE & COMMANDS

<https://github.com/zakiya/elnote>

https://www.evernote.com/l/AG2yQwu_m4JI6aVtDHJ4GLSOWJtyAYTG6E8



INSTALL NODE AND NPM

- nodejs.org for Node installer
- Go to npmjs.com & click documentation for instructions to update npm to latest



CREATE A PACKAGE

.....

1. Install package

```
$ cd [themeroor]
```

```
$ npm install [package] -save-dev
```

2. Visit NPM or Github page for documentation

3. Add script to package.json.

```
description: the best build tool in the
"main": "index.js",
"scripts": {
  "sassitup": "node-sass --source-map true",
  "prefixit": "postcss --use autoprefixer",
  "squishcss": "minify --output production",
  "squishjs": "minify --output production",
  "obeymejs": "jshint js/*",
  "obeymecss": "sass-lint --config .sass-
  "taadaa": "watch 'npm run sassitup && n
obeymecss' sass.js",

  "makesvg": "svg-sprite-generate -d svg
  "livereload": "live-reload --port 9091
```




AGENDA

- node-sass
- autoprefixer
- watch
- minify
- svg generator
- live reload
- susy, breakpoint
- linter
- terminal notifier

NODE-SASS

Goal: Compiles Sass

```
"sassitup": "node-sass --source-map true --include-path sass  
sass/styles.scss css/styles.css"
```

- **source map** - generates a file that helps browsers debug
- **include-path**
- paths are always relative to package.json

AUTOPREFIXER, POSTCSS-CLI

Goal: Add vendor prefixes to CSS

```
"prefixit": "postcss --use autoprefixer --autoprefixer.browsers  
  'last 2 versions, iOS 8' --replace css/styles.css"
```

- `autoprefixer.browsers` 'last 2 versions, iOS8' - define versions
- `replace` - name of resulting file
- See:
 - <https://github.com/postcss/autoprefixer>
 - <https://github.com/ai/browserslist#queries>

MINIFIER

Goal: Minify CSS and Javascript

```
"squishcss": "minify --output production/styles.min.css css/styles.css",  
"squishjs": "minify --output production/scripts.min.js js/scripts.js"
```

- **output** - name of source file
- may need to create source file before running script the first time.
- Alternative: uglify

WATCH

Goal: Execute Script on change

```
"taadaa": "watch 'npm run sassitup && npm run prefixit && npm run squishcss  
&& npm run squishjs' sass js"
```

- other packages do the same like
- watch '[commands to run]' folder1 folder2 folder3
- Alternatives: onchange dirwatch, nodemon
- You can also run tasks asynchronously or use pre- and post-hooks
- See | > &

SUSY, BREAKPOINT-SASS

Goal: Use Sass libraries

- No script to add, just an `@import` at the top of the `styles.css`
- NPM packages also exist for singularity, compass, and mq

```
@import '../node_modules/breakpoint-sass/stylesheets/breakpoint';  
@import '../node_modules/susy/sass/susy';
```

JSHINT

Goal: Enforce Javascript code standards

```
"obeymejs": "jshint js/*",
```

- consider using editor's linter instead

SASS-LINT

Goal: Enforce Sass code standards

```
"obeymecss": "sass-lint --config .sass-lint.yml 'sass/*.scss' --verbose  
  --no-exit"
```

- Note: sass-lint is not scss-lint (has ruby dependency)
- sample config: `node_modules/sass-lint/docs/sass-lint.yml`
- see config <https://github.com/sasstools/sass-lint/tree/master>

```
#####  
## Sample Sass Lint File  
#####  
# Linter Options  
options:  
  # Don't merge default rules  
  merge-default-rules: false  
  # Set the formatter to 'html'  
  formatter: stylings  
  # Output file instead of logging results  
  #output-file: 'linters/sass-lint.html'
```

LIVE-RELOAD

Goal: Refresh browser when file or directory is changed

```
"reload": "live-reload --port 9091 production/",
```

- run this script once. Doesn't need to be added to watch script
- Add to .html `<script src="//localhost:9091"></script>`
- can also use gui + browser extension

SVG-SPRITE-GENERATOR

Goal: Generate SVG sprite from source files

```
"makesvg": "svg-sprite-generate -d svg -o svg/sprite.svg",
```

- see <https://css-tricks.com/svg-sprites-use-better-icon-fonts/>
- only run once, or when new file is added.
- d - source directory
- o - output file

NOTIFIER (NOT NPM)

Goal: Desktop notifications (OS X)

```
"taadaa": "watch 'npm run sassitup && npm run prefixit && npm run squishcss  
&& npm run squishjs && npm run obeymejs && npm run obeymecss &&  
terminal-notifier -message 'complete' sass js',
```

- Couldn't find a notify package with a NPM binary
- <https://github.com/julienXX/terminal-notifier>
 - Install with brew or gem install
 - you can also use && echo 'complete'

OTHER THING THAT CAN BE DONE WITH NPM

- Image minifier & optimizers
- sassdoc (self-documents sass files)
- filesize (can notify when packages get to big)
- browser-sync (reloads browsers across devices)



PREPARE FOR EXPORT

.....

- gitignore
 - /node_modules
 - /production (?)
 - css/styles.css.map
- Write a thorough README.md



USING PACKAGE

- cd to the root of your Drupal theme/project
- `$ npm update` (updates global npm)
- `$ npm install` (one time)
- `$ npm run {{nameofscript}}`

CONVINCED?

RESOURCES

RESOURCES – SHORTCUTS

- -s silences output from NPM
- -i install
- -u use
- pre- and post- hooks

RESOURCES – LINKS

- <https://docs.npmjs.com/getting-started/installing-node>
- <http://davidthec Clark.com/its-time-for-everyone-to-learn-about-postcss/>
- <https://css-tricks.com/why-npm-scripts/>
- <https://medium.freecodecamp.com/why-i-left-gulp-and-grunt-for-npm-scripts-3d6853dd22b8#.xc43pleqn>
- http://substack.net/task_automation_with_npm_run
- <https://medium.com/@iamryanyu/svg-sprite-workflow-that-works-f5609d4d6144#.gyfnpbhes>
- <https://sarasoueidan.com/blog/icon-fonts-to-svg/>
- <http://blog.keithcirkel.co.uk/how-to-use-npm-as-a-build-tool/>
- <http://blog.keithcirkel.co.uk/why-we-should-stop-using-grunt/>
- <http://blog.modulus.io/using-npm-scripts-to-build-asset-pipeline>

NICE LOBSTER'S POOFIE EMPORIUM

etsy.com/shop/nicalorber

QUESTIONS?