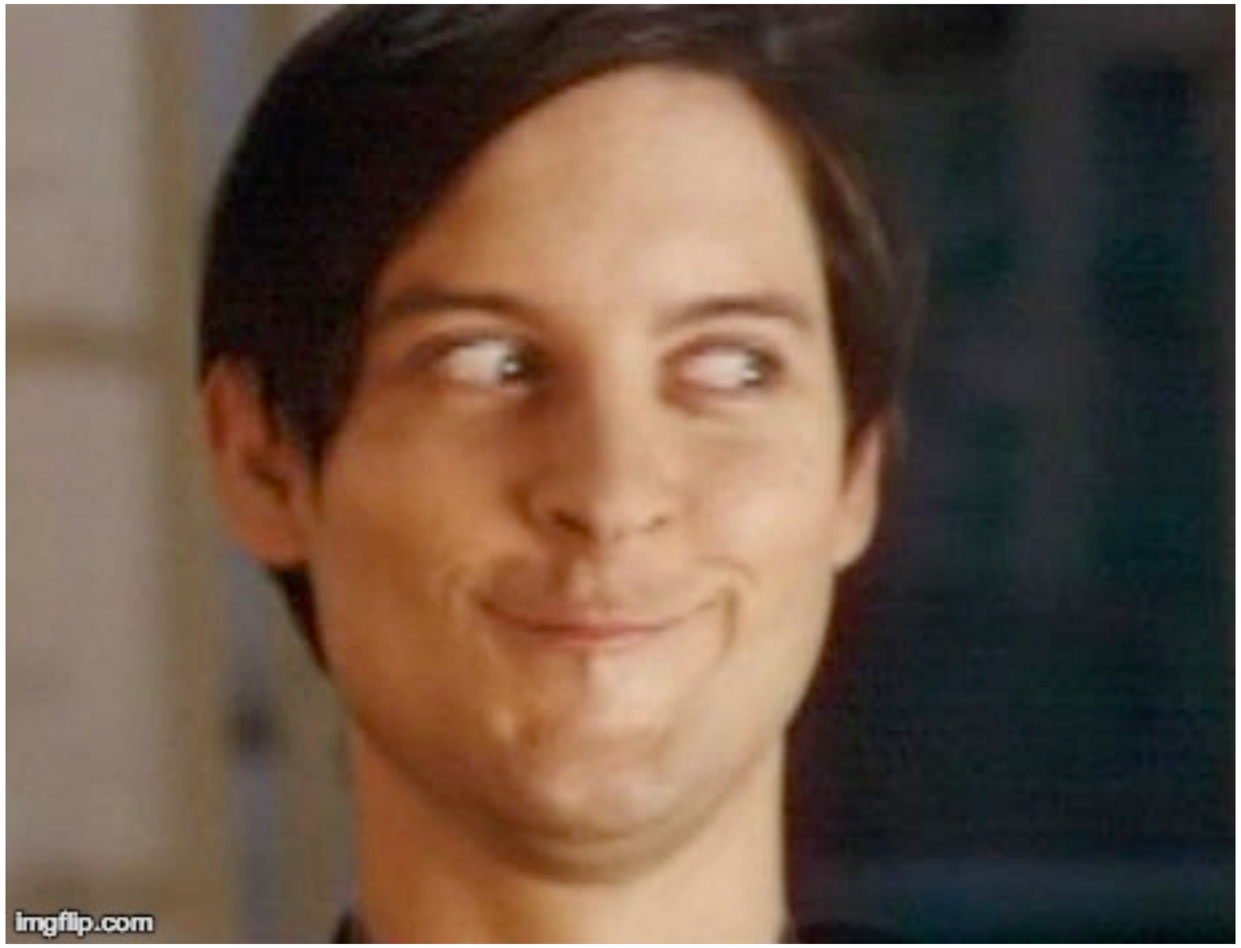
Assets Build Automation

Marconi Moreto

@marconimjr

Story time…



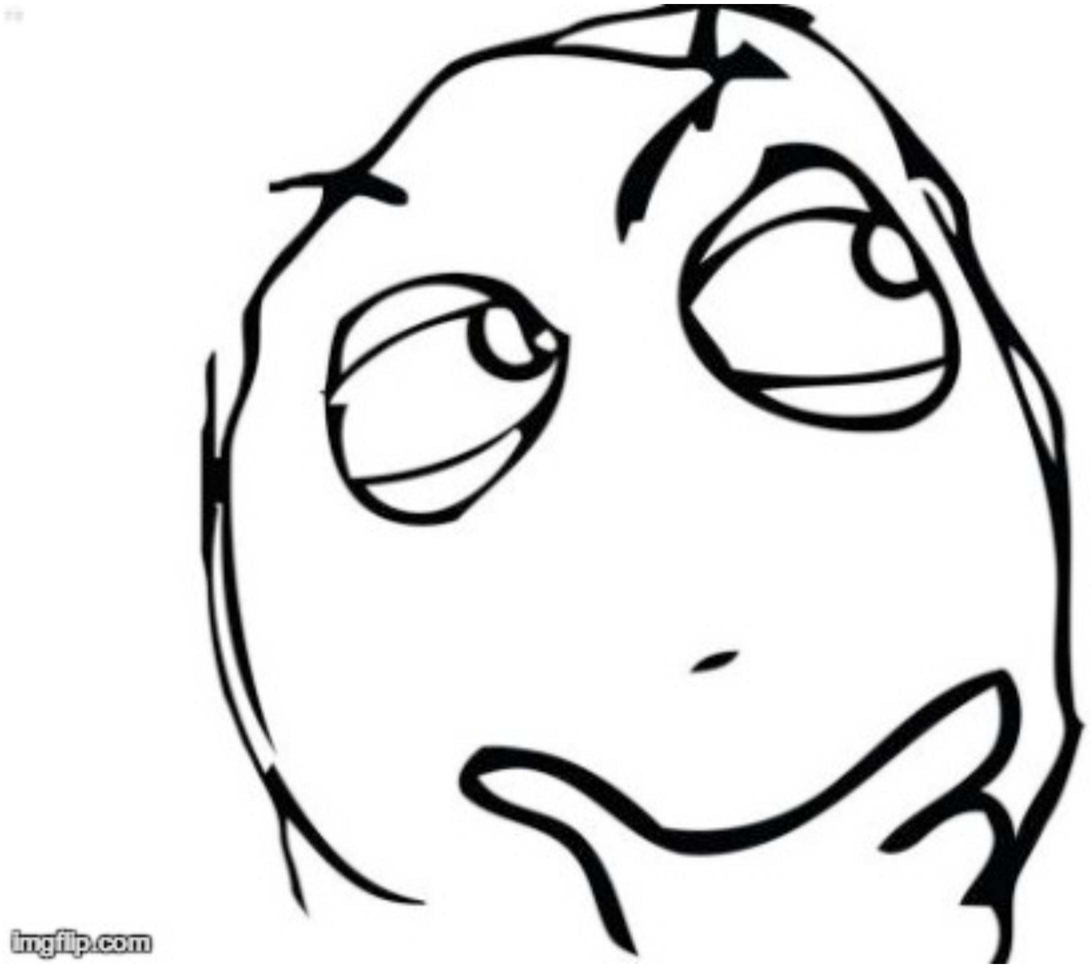
Classic way:

```html
<!doctype html>
<html>
<head>
  <title>My Awesome WebApp</title>
  <link href="/static/css/somegrid.css" type="text/css" rel="stylesheet"/>
  <link href="/static/css/plugins.css" type="text/css" rel="stylesheet"/>
  <link href="/static/css/styles.css" type="text/css" rel="stylesheet"/>
</head>
<body>

  ...

  <script src="/static/js/jquery.js" type="text/javascript"></script>
  <script src="/static/js/plugins.js" type="text/javascript"></script>
  <script src="/static/js/app.js" type="text/javascript"></script>
</body>
</html>
```

Number of requests: 7



imgflip.com

Concatenate everything:

```html
<!doctype html>
<html>
<head>
  <title>My Awesome WebApp</title>
  <link href="/static/css/app.css" type="text/css" rel="stylesheet"/>
</head>
<body>

  ...

  <script src="/static/js/app.js" type="text/javascript"></script>
</body>
</html>
```

Number of requests: 3

Quick peak 1/2

app.css

```css
body {
  padding: 0;
  margin: 0;
}

#notification {
  width: 100%;
  font-size: 18px;
  text-align: center;
  padding: 15px 0;
}

#notification.success {
  background: #1abc9c;
}

#notification.error {
  background: #e74c3c;
}
```

Size: 277 bytes

app.js

```javascript
/**
 * Basic notification class
 */
function Notification(msg, type) {
  this.msg = msg;
  this.type = typeof type !== 'undefined' ? type : 'success';
}

Notification.prototype.show = function() {
  // build notification
  var notif = document.createElement('div');
  notif.id = 'notification';
  notif.className = this.type;
  notif.innerHTML = this.msg;

  // prepend notificaiton
  var doc = document.getElementsByTagName("body")[0];
  doc.insertBefore(notif, doc.firstChild);
};

$(function() {
  new Notification("Welcome!").show();
})();
```

Size: 543 bytes

MINIFY EVERYTHING!
H HD
HISTORY.COM
imgflip.com

Minify everything:

```html
<!doctype html>
<html>
<head>
  <title>My Awesome WebApp</title>
  <link href="/static/css/app.min.css" type="text/css" rel="stylesheet"/>
</head>
<body>

  ...

  <script src="/static/js/app.min.js" type="text/javascript"></script>
</body>
</html>
```

✓ Number of requests: 3

✓ Faster loading time

Quick peak 2/2

app.min.css

```
body{padding:0;margin:0}
#notification{width:
100%;font-size:18px;text-
align:center;padding:15px
0}
#notification.success{backg
round:#1abc9c}
#notification.error{backgro
und:#e74c3c}
```

~~Size: 277 bytes~~

Size: 177 bytes

app.min.js

```
function Notification(a,b){this.msg=a,this.type="undefined"!
=typeof b?b:"success"}Notification.prototype.show=function()
{var
a=document.createElement("div");a.id="notification",a.classNam
e=this.type,a.innerHTML=this.msg;var
b=document.getElementsByTagName("body")
[0];b.insertBefore(a,b.firstChild)},$(function(){new
Notification("Welcome!").show()})();
```

~~Size: 543 bytes~~

Size: 352 bytes

COOL STORY BRO
imgflip.com

Why do all these?

SPEED
LIMIT
WHATEVER
SAVE LIVES
SPEED MATTERS
imgflip.com

DOING MANUALLY
IS A DISASTER
imgflip.com

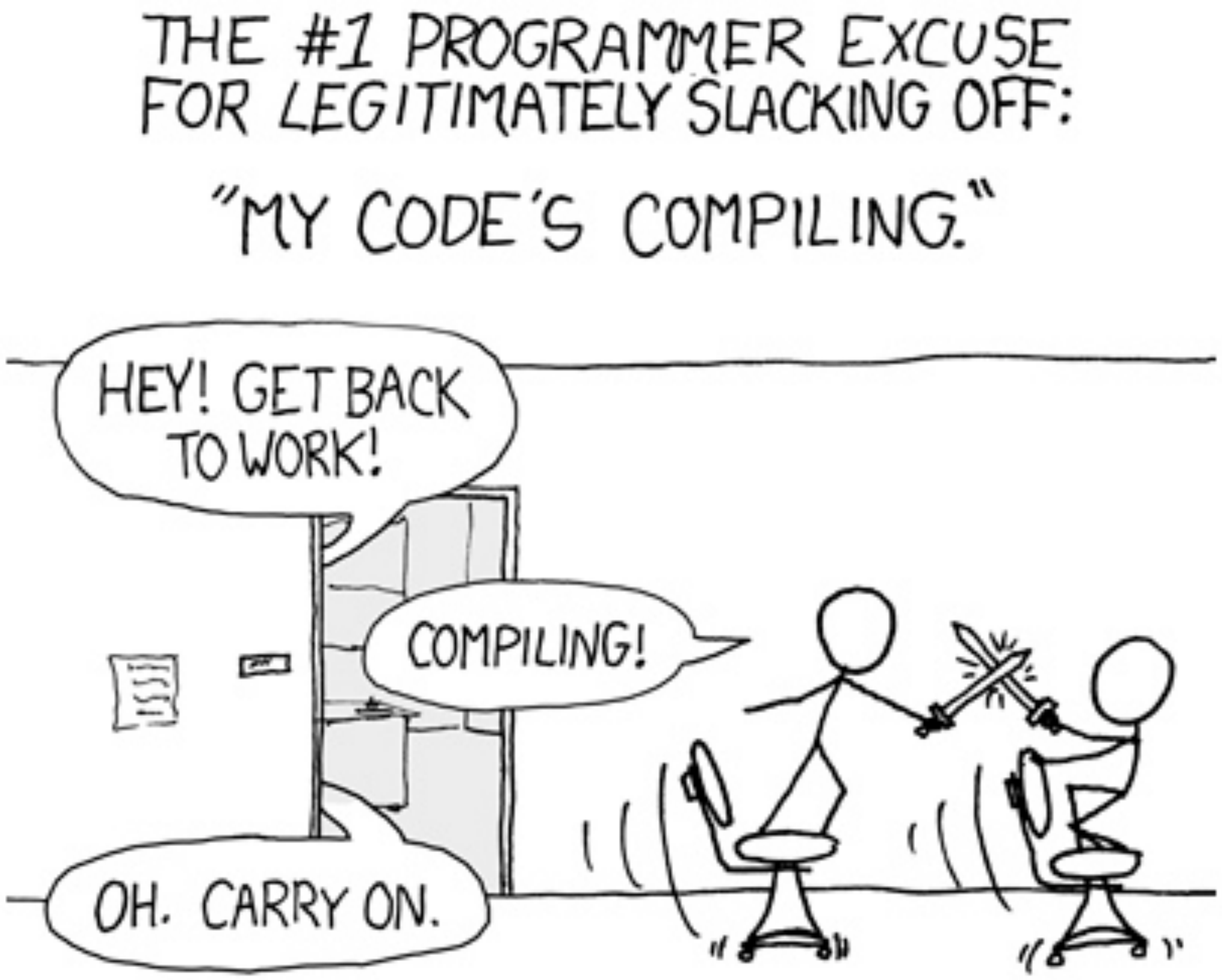
AUTOMATE EVERYTHING
imgflip.com

But why do we automate?

BECAUSE DEVELOPERS
ARE LAZY!
imgflip.com

But really, why do we automate?

Focus on fun stuff!

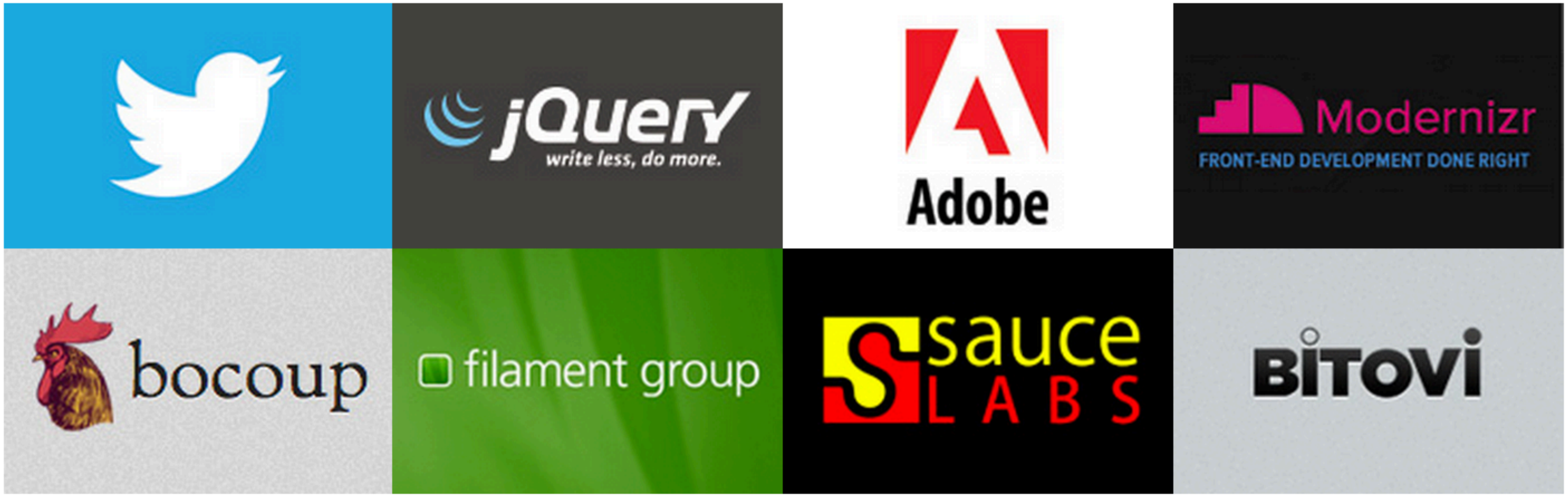


The JavaScript Task Runner

Things to love about Grunt:

- ✓ Node
- ✓ JS/Coffee + JSON
- ✓ Focus on web development
- ✓ Over 2,294 plugins!

Who's using it?



And some more: Wordpress, Opera, etc.

Getting started:

Installation:

```
$ npm install -g grunt-cli
```

Requirements:

```
- package.json
- Gruntfile.(js|coffee)
```

package.json

```
{
  "name": "MyAwesomeApp",
  "version": "0.0.1",
  "private": true
}
```

Install dependencies:

```
$ npm install grunt --save
$ npm install grunt-contrib-uglify --save
$ npm install grunt-contrib-cssmin --save

...

$ cat package.json
{
  "name": "MyAwesomeApp",
  "version": "0.0.1",
  "private": true,
  "dependencies": {
    "grunt": "~0.4.2",
    "grunt-contrib-uglify": "~0.3.2",
    "grunt-contrib-cssmin": "~0.8.0"
  }
}
```

CSS Minification

Gruntfile.js

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    cssmin: {
      styles: {
        files: {
          'static/css/app.min.css': ['src/css/*.css']
        }
      }
    }
  });
  grunt.loadNpmTasks('grunt-contrib-cssmin');
  grunt.registerTask('default', ['cssmin']);
};
```

Lets run it!

```
$ grunt
```

JS Minification

Gruntfile.js:

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    cssmin: {
      styles: {
        files: {
          'static/css/app.min.css': ['src/css/*.css']
        }
      }
    },
    uglify: {
      scripts: {
        files: {
          'static/js/app.min.js': [
            'src/js/jquery.js',
            'src/js/plugins.js',
            'src/js/app.js'
          ]
        }
      }
    }
  });
  grunt.loadNpmTasks('grunt-contrib-cssmin');
  grunt.loadNpmTasks('grunt-contrib-uglify');
  grunt.registerTask('default', ['cssmin']);
  grunt.registerTask('deploy', ['cssmin', 'uglify']);
};
```

Lets run it!

```
$ grunt deploy
```

Concatenate on development + minify on deploy

Install plugin:

```
$ npm install grunt-contrib-concat --save
```

Gruntfile.js:

```
module.exports = function(grunt) {
  grunt.initConfig({
    ...
    concat: {
      dist: {
        src: ['src/js/**/*.js'],
        dest: 'static/js/app.all.js'
      }
    },
    uglify: {
      scripts: {
        files: {
          'static/js/app.min.js': ['static/js/app.all.js']
        }
      }
    }
  });
  ...
  grunt.loadNpmTasks('grunt-contrib-concat');
  grunt.registerTask('default', ['cssmin', 'concat']);
  grunt.registerTask('deploy', ['cssmin', 'concat', 'uglify']);
};
```

Lets run it!

```
$ grunt deploy
```

What about SASS?

Updated project structure:

```
$ tree .
.
├── public
│   └── index.html
├── src
│   ├── Gruntfile.js
│   ├── js
│   │   ├── jquery.js
│   │   ├── app.js
│   │   └── plugins.js
│   ├── package.json
│   └── sass
│       ├── _plugins.scss
│       └── app.scss
└── static
    ├── css
    │   └── app.min.css
    └── js
        ├── app.all.js
        └── app.min.js
```

_plugins.scss

```
#notification {
  width: 100%;
  font-size: 18px;
  text-align: center;
  padding: 15px 0;

  &.success {
    background: #1abc9c;
  }

  &.error {
    background: #e74c3c;
  }
}
```

app.scss

```
body {
  padding: 0;
  margin: 0;
}

@import "plugins";
```

Install plugin:

```
$ npm install grunt-sass --save
```

Gruntfile.js:

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    sass: {
      dist: {
        options: {
          outputStyle: 'compressed'
        },
        files: {
          '../static/css/app.min.css': 'sass/app.scss'
        }
      }
    },
    concat: {
      ...
    },
    uglify: {
      ...
    }
  });
  grunt.loadNpmTasks('grunt-sass');
  grunt.loadNpmTasks('grunt-contrib-concat');
  grunt.loadNpmTasks('grunt-contrib-uglify');
  grunt.registerTask('default', ['sass', 'concat']);
  grunt.registerTask('deploy', ['sass', 'concat', 'uglify']);
};
```

Watching files

Install plugin:

```
$ npm install grunt-contrib-watch --save
```

Gruntfile.js:

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    ...
    watch: {
      sass: {
        files: ['sass/**/*.scss'],
        tasks: ['sass']
      },
      concat: {
        files: ['js/**/*.js'],
        tasks: ['concat']
      },
      uglify: {
        files: ['../static/js/app.all.js'],
        tasks: ['uglify']
      }
    }
  });
  ...
  grunt.loadNpmTasks("grunt-contrib-watch");
  ...
};
```

Watch changes:

```
$ grunt watch
```

index.html

```
<!doctype html>
<html>
<head>
  <title>My Awesome WebApp</title>
  <link href="/static/css/app.min.css" type="text/css" rel="stylesheet"/>
</head>
<body>

  ...

  <script src="/static/js/app.min.js" type="text/javascript"></script>
</body>
</html>
```

FINALLY
imgflip.com

BUT WE'RE NOT YET DONE
imgflip.com

app.scss

```scss
body {
  padding: 0;
  margin: 0;
}

#left, #right {
  width: 290px;
  height: 315px;
  float: left;
  margin-right: 10px;
}

#left {
  background: url('../images/left.jpg') no-repeat top left;
}

#right {
  background: url('../images/right.jpg') no-repeat top left;
}

@import "plugins";
```

index.html

```
<!doctype html>
<html>
<head>
  <title>My Awesome WebApp</title>
  <link href="/static/css/app.min.css" type="text/css" rel="stylesheet"/>
</head>
<body>

  ...

  <script src="/static/js/app.min.js" type="text/javascript"></script>
</body>
</html>
```

Number of requests: 5

Install plugin:

```
$ npm install grunt-image-embed --save
```

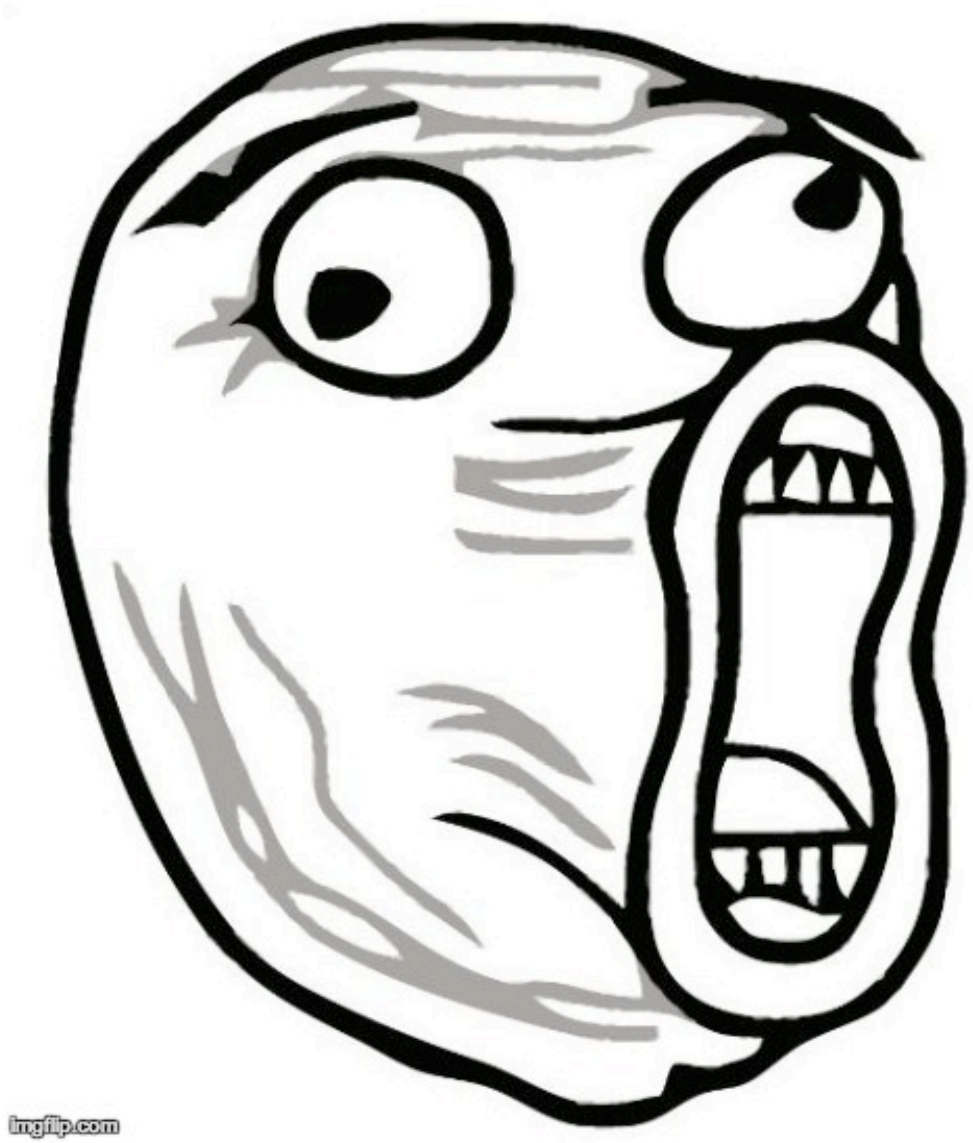
Gruntfile.js:

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    sass: {
      ...
    },
    imageEmbed: {
      dist: {
        src: ['../static/css/app.min.css'],
        dest: '../static/css/app.min.css',
        options: {deleteAfterEncoding : false}
      }
    },
    ...
    watch: {
      sass: {
        files: ['sass/**/*.scss'],
        tasks: ['sass', 'imageEmbed']
      },
      ...
    }
  });

  grunt.loadNpmTasks("grunt-contrib-watch");
  grunt.loadNpmTasks("grunt-image-embed");
  grunt.registerTask('default', ['sass', 'imageEmbed', 'concat']);
  grunt.registerTask('deploy', ['sass', 'imageEmbed', 'concat', 'uglify']);
};
```

Number of requests: 3

OMG!



Thank you

Marconi Moreto

@marconimjr

http://marconijr.com

https://github.com/marconi