JavaScript Build

Steve Hartzog

What is a JavaScript build?

We start with source…

… we perform a series of tasks on it

… then we output it for use.

Typical Build Tasks

1. **Linting** js formatting check
2. **Concatenation** merge multiple js files
3. **Minification** remove whitespace
4. **LESS Transpiling** convert less to css
5. JS Transpiling not used @ TC
6. Unit Test Runner not used @ TC
7. **Deploy Code** we curl it AEM
8. **Watcher** curls

What does the TC build look like?

Travel Channel Build Process

> grunt build

1. bowercopy
2. jshint
3. rjs ‘optimizer’
4. concat (*including* 3rd party libs)
5. uglify
6. less:compile
7. copy fonts

Then, we watch for changes...

Travel Channel Watcher

> grunt [default]

1. watches files*: jsp, js, less, css, jst
2. post to AEM

* No processing. No transpilation.

What is out there?

Build Options

Brunch Jan
2011
Grunt.js Mar
2012
Gulp.js Aug
2013
Broccoli Feb

Grunt.js

Currently in use across SNI
Tasks restricted to a json config
Designed for **SMALL** projects
Struggles with multi-step builds
I/O Bound
Project is practically dead...

Basic Gruntfile

```
grunt.initConfig({
  clean: {
    src: ['build/app.js', 'build/vendor.js']
  },
  copy: {
    files: [{
      src: 'build/app.js',
      dest: 'build/dist/app.js'
    }]
  }
  concat: {
    'build/app.js': ['build/vendors.js', 'build/app.js']
  }

  // ... other task configurations ...

});
grunt.registerTask('build', ['clean', 'concat', 'copy']);
```

Code: TC Gruntfile.js

What’s currently broken?

- Build step: rjs ‘optimizer’ (and AMD*)
  - has difficulty with CDNs
  - is VERY, VERY brittle
  - defeats the whole purpose of RequireJS
- Build takes 40-120 seconds
- In dev, less is compiled on the browser
  - 8-15 seconds before the page responds

* see last weeks overview of JavaScript Modules

What’s currently *missing*?

Build is not run by the Watcher

intermediate files are not cleaned up

mvn build sometimes required

No live reload

No unit tests

Gulp.js

Code over configuration
Small, reusable units
SIMPLE, elegant API
Not bound by I/O
Extremely FAST

Basic Gulpfile

```
gulp.task('default', function() {
  return gulp.src('src/styles/*.less')
    .pipe(less())
    .pipe(cssmin())
    .pipe(header(‘/* Scripps Networks Copyright 2015 */’))
    .pipe(gulp.dest('public/styles'));
});
```

Supports Advanced Builds

/build Build scripts in a
separate folder
/tasks Individual Tasks:
build.js basic build (orchestrator)
copy.js copy
dev.js basic dev setup
(orchestrator)
less.js transpile less to css
shell.js post to AEM
watch.js watch files, call *build*,
call *shell*
paths.js sets up variables to

Code: Gulp

Why move to GulpJS?

1. GulpJS project is very active
1. Builds are much Faster
1. Easier to update & maintain
1. Dev feedback loop is better
1. Plugin to run grunt tasks during the migration

Questions?