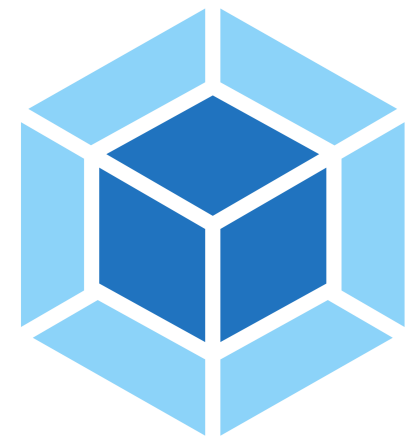# npm install gulp@next
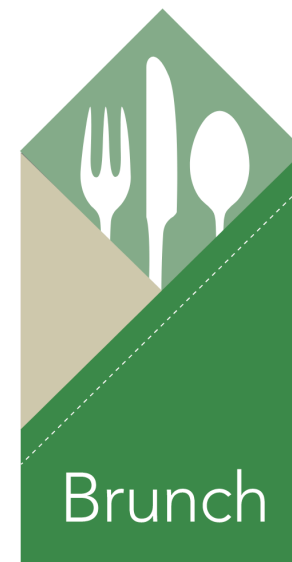
Blaine Bublitz - Lead Maintainer
Phoenix JavaScript (February 2018)

What is it?

Build System

Concerns

Task System

gulp.task(fn)

Orchestration

gulp.series(…fn)

gulp.parallel(…fn)

Asset Pipeline

gulp.src(globs)

gulp.dest(outDir)

.pipe(plugin)*

Watching

gulp.watch(globs, fn)

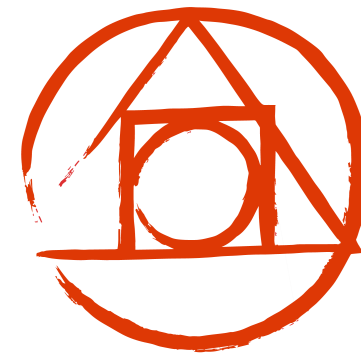***Provided by node.js streams**

Non-Concerns

Bundling

browserify

Language Compilation

TS

Gulp is just node - you can bring whatever tools you need for Bundling, Language Compilation, etc

BYOB (bring your own bundler)

BABEL

```
gulp.task('clean', function() {
  rimraf.sync('build');
});

gulp.task('html', ['clean'], function() {
  return gulp.src(['client/html/*.pug', '!client/html/_*.pug'])
    .pipe(pug())
    .pipe(gulp.dest('build/html'))
});

gulp.task('css', ['clean'], function() {
  return gulp.src('client/css/*.less')
    .pipe(sourcemaps.init({ loadMaps: true }))
    .pipe(less())
    .pipe(minifyCSS())
    .pipe(sourcemaps.write())
    .pipe(gulp.dest('build/css'))
});

gulp.task('js', ['clean'], function() {
  return rollup({
      input: 'client/js/index.js', sourcemap: true, format: 'iife'
    })
    .pipe(source('app.js'))
    .pipe(buffer())
    .pipe(sourcemaps.init({ loadMaps: true }))
    .pipe(babel({ presets: ['env'] }))
    .pipe(uglify())
    .pipe(sourcemaps.write('./'))
    .pipe(gulp.dest('build/js'))
});

gulp.task('default', ['html', 'css', 'js']);
```

```javascript
function clean(done) {
  rimraf('build', done);
}

function html(done) {
  pump([
    src(['client/html/*.pug', '!client/html/_*.pug']),
    pug(),
    dest('build/html')
  ], done);
}

function css(done) {
  pump([
    src('client/css/*.less', { sourcemaps: true }),
    less(),
    minifyCSS(),
    dest('build/css', { sourcemaps: true })
  ], done);
}

function js(done) {
  pump([
    rollup({
      input: 'client/js/index.js', sourcemap: true, format: 'iife'
    }),
    source('app.js'),
    buffer(),
    // Looking to replace this soon
    sourcemaps.init({ loadMaps: true }),
    babel({ presets: ['env'] }),
    uglify(),
    dest('build/js', { sourcemaps: './' })
  ], done);
}

export default series(clean, parallel(html, css, js));
```
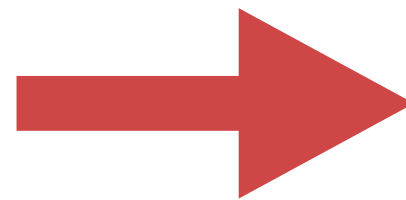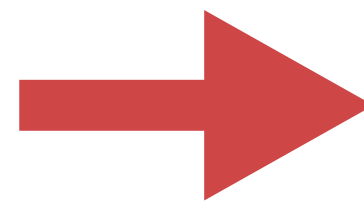
V4.0

What changed?

- No more sync tasks

```
gulp.task('clean', function() {
  rimraf.sync('build');
});
```

→

```
function clean(done) {
    rimraf('build', done);
}
```

- Named functions

- Avoid registering sub-tasks (clean, html, css, js)

```
gulp.task('clean', function() { ... });
gulp.task('html', ['clean'], function() { ... });
gulp.task('css', ['clean'], function() { ... });
gulp.task('js', ['clean'], function() { ... });
```

→

```
function clean(done) { ... }
function html(done) { ... }
function css(done) { ... }
function js(done) { ... }
```

Since we aren't registering sub-tasks, they don't each need a dependency on "clean"

What changed?

- pump instead of pipe*

```
gulp.task('html', ['clean'], function() {
  return gulp.src([
    'client/html/*.pug',
    '!client/html/_*.pug'
  ])
  .pipe(pug())
  .pipe(gulp.dest('build/html'))
});
```

```
function html(done) {
  pump([
    src([
      'client/html/*.pug',
      '!client/html/_*.pug'
    ]),
    pug(),
    dest('build/html')
  ], done);
}
```

- Integrated source map support

```
gulp.src('client/css/*.less')
  .pipe(sourcemaps.init({ loadMaps: true }))
```

```
.pipe(sourcemaps.write())
.pipe(gulp.dest('build/css'))
```

```
src('client/css/*.less', { sourcemaps: true })
```
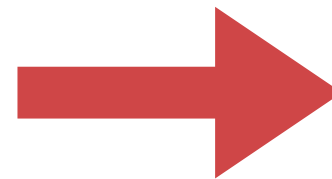
```
dest('build/css', { sourcemaps: true })
```

What changed?

- Task orchestration using series & parallel

```
gulp.task('js', ['clean'], function() { ... });
gulp.task('default', ['html', 'css', 'js']);
```

```
series(
  clean,
  parallel(html, css, js)
);
```

- Exports!*

```
gulp.task('clean', ...);
gulp.task('html', ...);
gulp.task('css', ...);
gulp.task('js', ...);
gulp.task('default', ...);
```

```
module.exports = {
  html, css, js, // Just an example
  default: series(
    clean,
    parallel(html, css, js)
  )
};
```

```
export { html, css, js }; // Just an example
export default series(
  clean,
  parallel(html, css, js)
);
```

***Not restricted to gulp 4
but requires gulp-cli**

New! gulp-cli

- Separate global install - npm install gulp-cli

- Much smaller

- Supports gulp v3 and v4

- Versioned separately

- New features!

  - Task metadata

  - Config files

  - New flags (--verify, --series)

Further reading:

Gulp Sips: New Command Line Interface

Task metadata

Better document tasks for consumers

- .description = "Documentation string"
  - .flags = { documentation: "object" }

Demo time!

Further reading:

Gulp Sips: Custom Task Metadata

New! Async completion

Previously, we only supported:

- Streams
- Promises (spotty)
- Callbacks

But now, we support:

- Streams
- Promises
- Event Emitters
- Child Processes
- Observables
- Callbacks
- Async/Await

My favorite demo :)

New! Incremental builds

Only process changes files

- Operates on any timestamp
- Most useful with gulp.watch()
- Fast because unchanged files aren't read

More demos!

New! Custom registries

Warning: Advanced feature

Share functionality or tasks across projects

I can deep dive here if the audience wants.

Final demo

Detailed information in the
Undertaker Docs for Custom Registries

Even more new things

- gulp.src() is a passthrough stream (no more gulp-add-src)

- gulp.symlink()

- gulp.tree()

- --tasks-json flag

- A real logger

- Gulp Sips blog posts

- Documentation website

- Probably more that I'm forgetting...

Official documentation for all this coming soon.
(Please help!)

Thanks!

https://opencollective.com/gulpjs

https://twitter.com/gulpjs

https://github.com/gulpjs

https://github.com/phated

Slides and Demos:
https://github.com/phated/talks

Spot the bug #1

```
gulp.task('clean', function() {
  rimraf('./build')
});

gulp.task('build', ['clean'], function() {
  return gulp.src('lots-of-files/*.txt')
    .pipe(gulp.dest('build'));
});

gulp.task('default', ['build']);
```

This is bug due to "clean" being treated as a sync task because nothing was returned and callback wasn't used.

Spot the bug #2

```javascript
gulp.task('clean', function(cb) {
  rimraf('./build', cb)
});

gulp.task('build', function() {
  return gulp.src('lots-of-files/*.txt')
    .pipe(gulp.dest('build'));
});

gulp.task('default', ['clean', 'build']);
```

This bug is due to "build" not depending on "clean" and the task dependencies of "default" running in parallel.

Spot the bug #3

```
/* Specific to gulp 4 */
var dirs = [
  'lots-of-files'
];

gulp.task('default', function() {
  var globs = dirs.map((dir) => path.join(__dirname, dir, '*.txt'));
  return gulp.src(globs)
    .pipe(gulp.dest('build'))
});
```

This bug is due to path.join and __dirname using \\ separator on Windows, which isn't a valid glob pattern.