



Florian Rapp

Gulp

Profil

- Freiberuflicher IT-Berater
- Doktorand Theoretische Physik
- Microsoft MVP für Visual C#
- CodeProject MVP

Build Automation

- Skripte
- Make
- Grunt
- Gulp

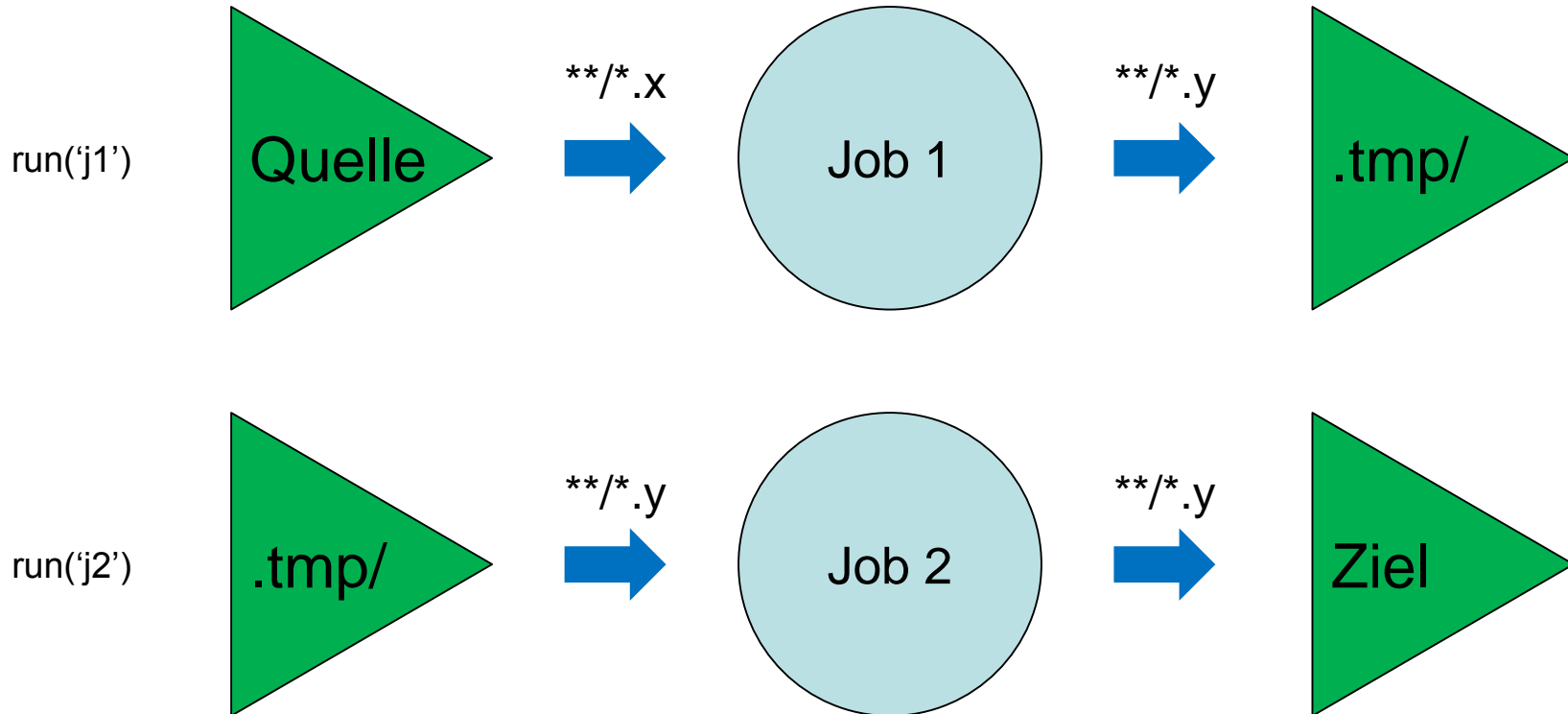


GRUNT

Vorteile von Grunt

- Task Runner
- Configuration-over-scripting
- Spezialisierte Tools
- Plugins
- Konsistenz

Ablauf





Warum nun gulp?

- Geschwindigkeit
- Code-over-Configuration
- Plugins + Module
- Elegante API
- Pipe Konzept

Installation

```
# Einmalig (global)
```

```
$ npm install -g gulp
```

```
# Pro Projekt
```

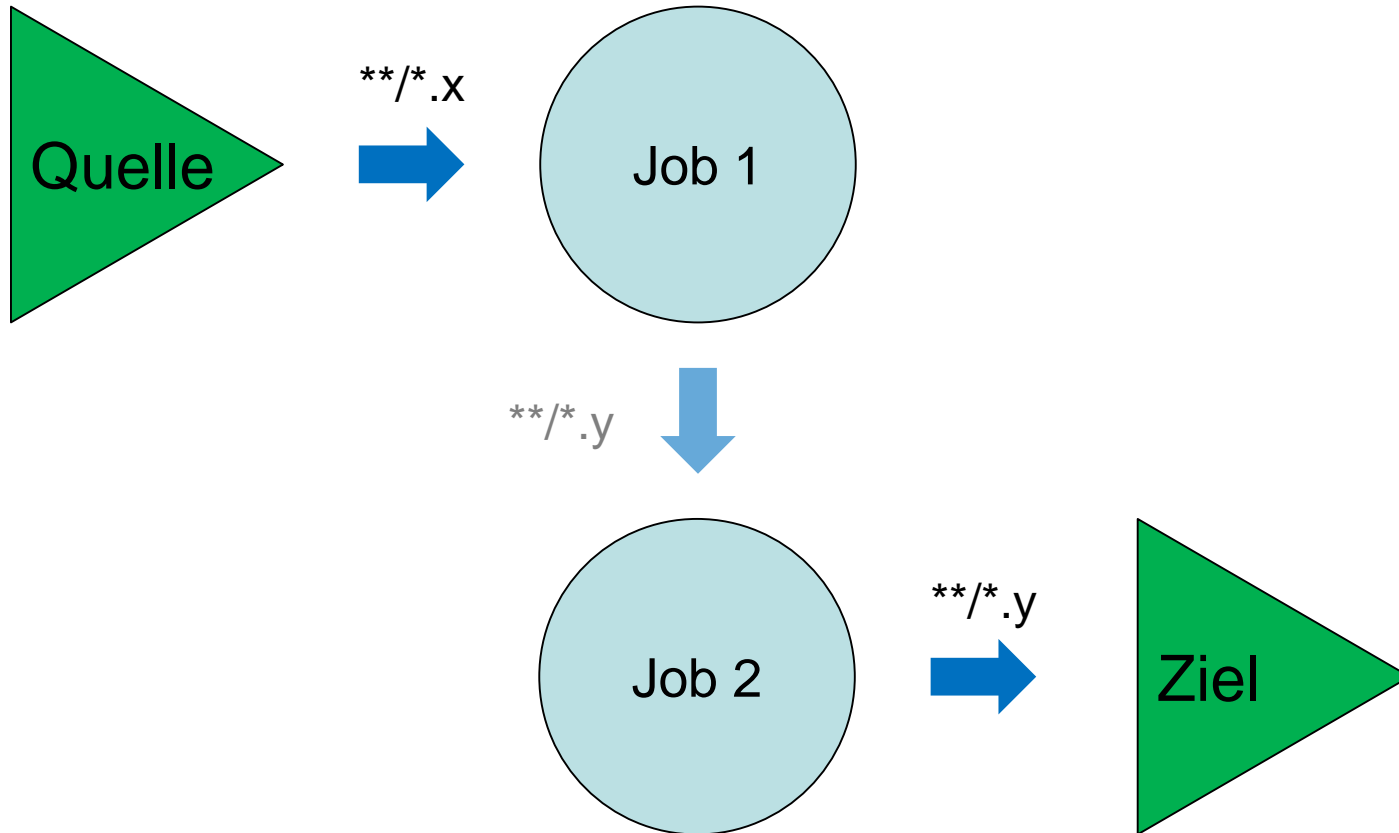
```
$ npm init # Nur ohne package.json nötig
```

```
$ npm install --save-dev gulp
```

Streaming

- Keine temporären Dateien
- Erhöhte Geschwindigkeit
- Direktere API
- Funktionaler Zugriff

Ablauf



Beispiel (gulpfile.js)

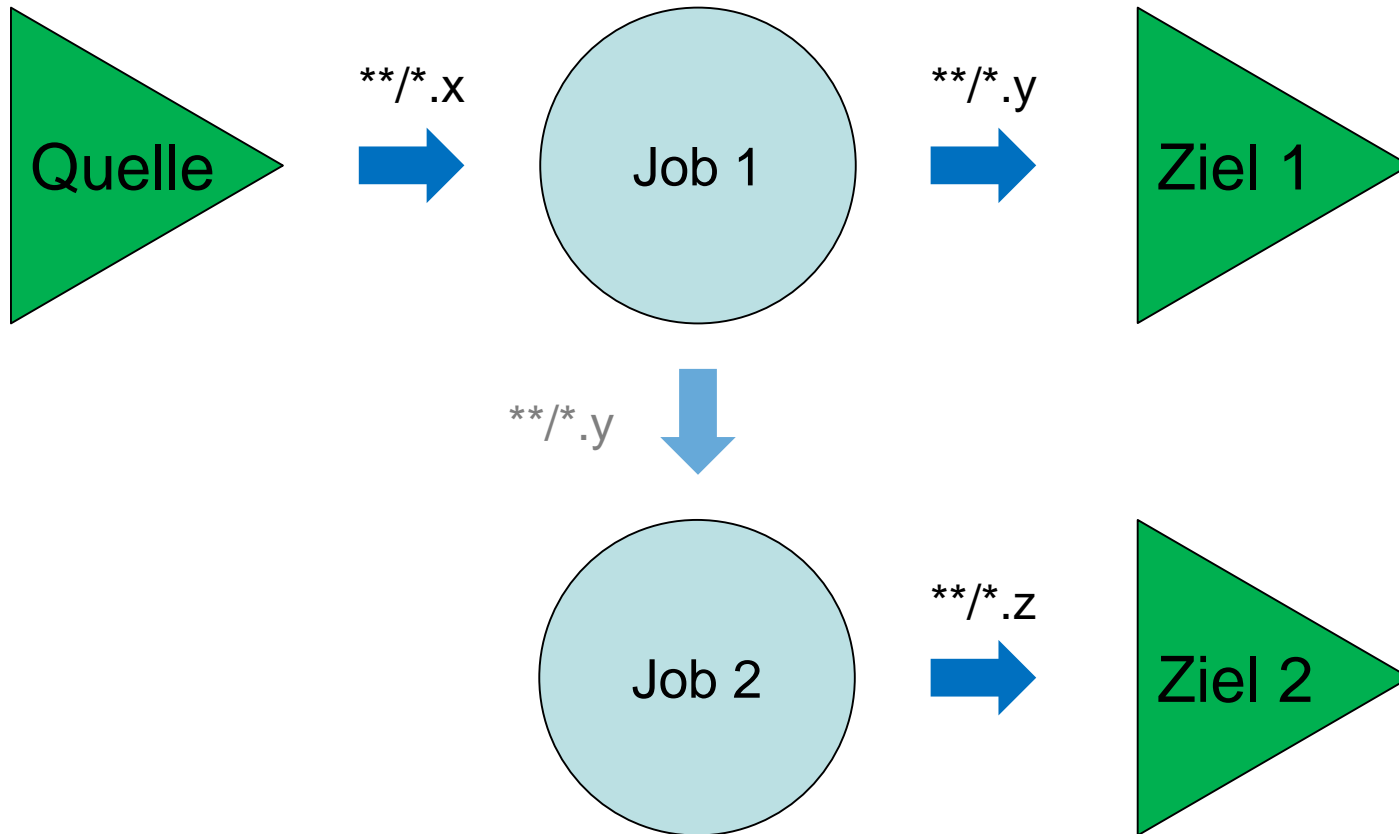
```
var gulp = require('gulp'),
    uglify = require('gulp-uglify');

gulp.task('minify', function() {
  gulp.src('js/app.js')
    .pipe(uglify())
    .pipe(gulp.dest('build'));
});
```

Quelle und Ziel

- `gulp.src()` lokalisiert Quelldateien
- Ein String oder Array von Strings, z.B.:
 - `js/app.js` oder `!js/app.js`
 - `js/*.js` oder `js/**/*.js`
 - `*.+(js|css)`
- `gulp.dest()` schreibt Stream

Multi Deploy



Beispiel

```
gulp.task('css', function() {  
  gulp.src('src/css/*.css')  
    .pipe(autoprefixer())  
    .pipe(gulp.dest('build/css/'))  
    .pipe(minifyCss())  
    .pipe(rename({extname: '.min.css'}))  
    .pipe(gulp.dest('build/css/'));  
});
```

Ausführen

```
$ gulp minify css
```

- Sucht: *gulpfile.js*
- Führt Task (*minify*) aus
- Mehrere Tasks möglich
- Status

Tasks

- Sind Build Ziele
- Spezieller Task: `default`
- Können Abhängigkeiten darstellen und besitzen
- Rückgabe des Streams im Task um Abwarten zu ermöglichen

Beispiel (obsolete)

```
gulp.task('default', ['clean'], function() {  
  gulp.start('styles', 'scripts');  
});
```

↑
Abhängigkeiten

Beispiel (Oct 2014)

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

```
gulp.task('clean', function() {  
  !return gulp.src('build', { read: false })  
    .pipe(clean());  
});
```



```
var clean = require('gulp-clean');
```

Plugins

- Installation über npm
- Einfache Erstellung
- Meist sehr klein
- Klare Verwendung

Plugin Auswahl (Allgemein)

- Zusammenfügen (`gulp-concat`)
- LiveReload (`gulp-livereload`)
- Meldungen (`gulp-notify`)
- Clean build (`del` oder `gulp-clean`)

Beispiel

```
var gulp = require('gulp'),
    concat = require('gulp-concat');

gulp.task('concat', function() {
    gulp.src('*.*js')
        .pipe(concat('concat.js'))
        .pipe(gulp.dest('build'));
});
```

Plugin Auswahl (CSS)

- SASS (`gulp-sass`)
- LESS (`gulp-less`)
- Autoprefixer (`gulp-autoprefixer`)
- CSS Minifier (`gulp-minify-css`)
- Bilderkomprimierung (`gulp-imagemin`)

Beispiel

```
gulp.task('images', function() {  
  return gulp.src('src/images/**/*')  
    .pipe(imagemin({  
      optimizationLevel: 3,  
      progressive: true,  
      interlaced: true }));  
  .pipe(gulp.dest('dist/assets/img'))  
  .pipe(notify({ message: 'Done!' }));  
});
```


Plugin Auswahl (JS)

- JSHint (`gulp-jshint`)
- Uglify (`gulp-uglify`)
- Style Guide (`gulp-jscs`)
- Dokumentation (`gulp-jsdoc`)

Caching und mehr

- Pakete für Caching
 - gulp-cached
 - gulp-remember
- Nur geänderte Dateien
 - gulp-changed

Watches

- Build-on-change
- Überwacht Dateien
- Tasks bei Änderung ausführen

Beispiel

```
gulp.task('watch', function() {  
  gulp.watch('./js/src/*.js', ['js']);  
});
```

↑
zu startende Tasks

LiveReload

- Plugin
- Ergänzung zu Watches
- Benötigt Browser-Plugin
- Alternativ: BrowserSync
 - Ohne Browser Plugin
 - Mehr Funktionalität

Beispiel

```
gulp.task('watch', function() {  
  // Rebuild Tasks  
  livereload.listen();  
  gulp.watch(['build/**'])  
    .on('change', livereload.changed);  
});
```

LiveReload Server starten



Neuladen bei Änderungen



Eigene Plugins

- Einzigartig
- Fokussiert
- Unabhängig von Gulp
- Benutze `PluginError` von `gulp-util`
- Verwende Tag `gulpplugin`

Boilerplate

```
var through = require('through2'),
    gutil = require('gulp-util'),
    PluginError = gutil.PluginError;

module.exports = function() {
  return through.obj(function(file, enc,
    cb) {
    /* ... */
  });
};
```


Vergleich

Grunt

- JSON Konfig.
- Sehr viele Plugins
- Mächtige Plugins
- Wrapper notwendig
- Dateibasiert

Gulp

- JS Code
- Viele Plugins
- Elementare Plugins
- Node modules
- Streambasiert

Beispielkonfiguration

```
grunt.initConfig({
  less: {
    development: {
      files: {
        "build/tmp/app.css": "assets/app.less"
      }
    }
  },
  autoprefixer: {
    options: {
      browsers: ['last 2 version', 'ie 8', 'ie 9']
    },
    multiple_files: {
      expand: true,
      flatten: true,
      src: 'build/tmp/app.css',
      dest: 'build/'
    }
  }
});
```

```
grunt.loadNpmTasks('grunt-contrib-less');
grunt.loadNpmTasks('grunt-autoprefixer');
```

```
grunt.registerTask('css', ['less', 'autoprefixer']);
```

```
var gulp = require('gulp'),
    less = require('gulp-less'),
    autoprefixer = require('gulp-autoprefixer');

gulp.task('css', function () {
  gulp.src('assets/app.less')
    .pipe(less())
    .pipe(autoprefixer('last 2 version', 'ie 8', 'ie 9'))
    .pipe(gulp.dest('build'));
});
```

Speedtest: SASS

```
05:33:51 🐙 {master} < ~/Dropbox/dropbox-htdocs
↳ $ grunt watch
Running "watch" task
Waiting...OK
>> File "scss/_buttons.scss" changed.

Running "sass:dev" (sass) task
File "css/kickoff-old-ie.css" created.
File "css/kickoff.css" created.

Running "sass:styleguide" (sass) task
File "css/styleguide.css" created.

Done, without errors.
Completed in 2.348s - Waiting...
```

Grunt: Sass
2.348 seconds

```
09:43:24 🐙 {gulp} < ~/Dropbox/dropbox-htdocs/g
↳ $ gulp
[gulp] Using file /Users/zmartineau/Dropbox/dro
[gulp] Working directory changed to /Users/zmar
[gulp] Running 'default'...
[gulp] Running 'lint'...
[gulp] Running 'sass'...
[gulp] Running 'scripts'...
[gulp] Finished 'default' in 20 ms
[gulp] Finished 'lint' in 115 ms
[gulp] Finished 'scripts' in 254 ms
[gulp] Finished 'sass' in 1.27 s
[gulp] Running 'sass'...
[gulp] Finished 'sass' in 1.27 s
```

Gulp: Sass
1.27 seconds

Speedtest: Uglify

```
06:40:04 🐼 {master} < ~/Dropbox/dropbox-htdocs  
↳ $ grunt watch  
Running "watch" task  
Waiting...OK  
>> File "js/script.js" changed.  
  
Running "uglify:js" (uglify) task  
Source Map "js/dist/app.min.js.map" created.  
File "js/dist/app.min.js" created.  
  
Done, without errors.  
Completed in 0.750s - Waiting...
```

Grunt: Javascript
750 milliseconds

```
09:44:05 🐼 {gulp} < ~/Dropbox/dropbox-htdocs/g  
↳ $ gulp  
[gulp] Using file /Users/zmartineau/Dropbox/dro  
[gulp] Working directory changed to /Users/zmar  
[gulp] Running 'default'...  
[gulp] Running 'lint'...  
[gulp] Running 'sass'...  
[gulp] Running 'scripts'...  
[gulp] Finished 'default' in 20 ms  
[gulp] Finished 'lint' in 125 ms  
[gulp] Finished 'scripts' in 258 ms  
[gulp] Finished 'sass' in 1.3 s  
[gulp] Running 'lint'...  
[gulp] Running 'scripts'...  
[gulp] Finished 'lint' in 49 ms  
[gulp] Finished 'scripts' in 100 ms
```

Gulp: Javascript
100 milliseconds

Alles auf einen Blick

1. `gulp.task(name, fn)`
2. `gulp.src(glob)`
3. `gulp.dest(folder)`
4. `gulp.watch(glob, fn)`
5. `gulp.start(tasks...)`

Fazit

- Sehr schnell
- Elegant
- Erweiterbar
- Einfach

Vielen Dank!



@FlorianRappl



<http://florian-rappl.de>

<https://github.com/FlorianRappl/GulpSamples>