



Breizh C@mp  
Mix de technologies



# GULP FICTION

builder, tester, livrer... sans tomber dans les



Matthieu Bréchet - @mbrechet  
Loïc Truchot - @n\_a\_n35





# Présentation des conférenciers



**Matthieu Bréchet**

Architecte Web

@mbréchet



**Loïc Truchot**

Lead Dev Web

@n\_a\_n35





# Introduction au sujet

- Pourquoi un environnement de *build* ?
  - un moteur de production: automatiser les tâches répétitives de construction de l'application
  - pratique dans tous les cas, indispensable en Agilité
- Lequel choisir ?
  - Plusieurs environnements très populaires: Make, Ant, Maven, Gradle, Grunt...
  - Pas de compétition: le choix se fait selon les besoins



# Notre application et notre besoin

- Une application full stack JavaScript
  - Notre App: la “tarantino-thèque” (mini-vidéothèque)
  - Contraintes spécifiques :
    - linter, combiner, minifier les fichiers JS, HTML, CSS, fonts, etc.
    - compiler le sass, compresser les images/sprites
    - déploiement continu sur IC et livraisons régulières





# Notre choix : Gulp

- Le choix de Gulp

- Maven sur notre projet VOD
- Grunt VS Gulp : pas de gagnant



- Gulp - le moteur de production “Node friendly”
- envie d’explorer les nouveautés sans oublier nos contraintes
- envie de coder plutôt que configurer



MAVEN ?



# Un build sous Maven

- Pour faire quoi ?

- vérifier
- builder
- tester
- packager
- déployer...





# Un build sous Maven

- Faisable oui, mais avec des contraintes
  - performance
  - maintenabilité
  - environnement



**C'EST À  
UNE DEMI  
HEURE**

**D'ICI J'Y SUIS  
DANS**

**IC  
MINUTES**



**Gulp ?**



# Installer Gulp pour builder

- Comment installer Gulp ?

- NodeJS et NPM : <https://nodejs.org/>
- Installation globale

```
> npm install -g gulp
```



- gestion des plugins

- le fichier [package.json](#)
- installation et sauvegarde

```
> npm init
```

```
> npm install --save gulp
```



# Gulp fonctionnement simple

- 1 fichier de description :  
le gulpfile
- 3 méthodes principales
  - task, src et dest
- une grande quantité de plugins disponibles
  - environ 1500
  - mais aussi les plugins NPM

<http://gulpjs.com/plugins/>

```
var gulp = require('gulp');
var uglify = require('gulp-uglify');
var concat = require('gulp-concat');
var del = require('del');

var config = {
  sourceDir: '../src',
  publicDir: '../client'
};

gulp.task('clean', function () {
  return function (done) {
    del(config.publicDir, {
      force: true,
      dot: true
    }, done);
  };
});

gulp.task('js-build',['clean'],function(){
  return gulp.src(config.sourceDir+'/js/*')
  .pipe(concat('scripts.js'))
  .pipe(uglify({mangle:false}))
  .pipe(gulp.dest(config.publicDir+'/js'));
});

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



# Gulp les bonnes pratiques

- 1 fichier par action

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

module.exports = function () {
    return gulp.src(global.dir.sources + '/js/*')
        .pipe(concat('scripts.js'))
        .pipe(uglify({
            mangle:false
        }))
        .pipe(gulp.dest(global.dir.client + '/js'));
};

gulp.task('js-build', require('./task/js-build'));
```

- Un répertoire spécifique pour le build
- le plugin run-sequence



# Lancer un build gulp !

- Une commande : > `gulp`
- Pour une tâche précise : > `gulp <task>`





# Aller plus loin avec Gulp 1/2

- En quoi Gulp est-il “node friendly” ?

- globs & vinyls
- streams & pipe
- plugin-gulp et NPM
- asynchronicité,  
parallélisme

```
var gulp = require('gulp');
var replace = require('gulp-replace');
var ftp = require('vinyl-ftp');
```

```
//FTP connexion
var connexionFTP = ftp.create({
  host:      'gulpfiction.com',
  user:      'VincentVega',
  password: 'Ezechiel35'
});

//Globs
var sourcesForRelease = [
  //included files
  global.dir.client + '**/*.js',
  global.dir.client + '**/*.html',
  global.dir.client + '**/*.css',
  //excluded file/folder are prepend by a "!"
  '!' + global.dir.client + '/devTools.js',
  '!' + global.dir.client + '/devToolsStyles.css'
];

gulp.src(sourcesForRelease)
  //Stream management
  .pipe(replace(/\$\{version\}/g, global.version))
  .pipe(gulp.dest(global.dir.release))
  .pipe(connexionFTP.dest('/public'));
```



# Aller plus loin avec Gulp 2/2

- Le watch `gulp.task('watch-webfiles', require('./tasks/watch-webfiles'));`
  - compilation de html : la tâche watch-webfiles

```
var gulp = require('gulp');
var browserSync = require('browser-sync');
var reload = browserSync.reload;

module.exports = function () {

    // Load Browser Sync server with proxy conf
    browserSync({
        port:9000, ui:{ port:9001 },
        files: ['./../client/**/*'], proxy:'localhost:3000'
    });

    // Watch files for changes & reload
    gulp.watch([global.dir.sources + '**/*.html'], ['html', reload]);
};
```

- et pour les scss ?



# Conclusion

- La promesse est tenue !
- Gulp aujourd'hui
  - innombrables utilisateurs
  - activité encore quotidienne sur le repo github
  - faiblesses actuelles : erreurs, sourcemaps, site et docs - objectifs de la 4.0 !
- Questions ?





# Annexe

- notre biblio

- <http://gulpjs.com>
- <http://nodejs.org>
- <http://gulpfiction.divshot.io/>
- <https://medium.com/@contrahacks/gulp-3828e8126466>

- accéder à nos sources

- demo : <http://github.com/mbrechet/gulpfiction>
- présentation : <https://goo.gl/VUj5Oy>