

Les outils du développeur Web

Node Package Manager (npm)

Initialisation npm du projet

Création du fichier package.json bash `npm init`

Installation d'un paquet :

```
npm install [options] paquet
```

Options: * -g: installation globale * --save-dev: ajout dans package.json

Exemples : Installation globale d'un paquet comme super admin:

```
npm install -g gulp
```

Installation locale d'un paquet:

```
npm install --save-dev gulp-connect
```

Installer les dépendances du projet (package.json):

```
npm install

gulp@3.9.0 node_modules/gulp
  pretty-hrtime@1.0.0
  interpret@0.6.5
  ...

gulp-connect@2.2.0 node_modules/gulp-connect
  connect-livereload@0.3.2
  event-stream@3.1.7 (...)
  ...
```

Ressources : * <https://nodejs.org> * <https://docs.npmjs.com/cli/install>

NPM Lazy

Proxy/cache à déployer sur le réseau local de l'entreprise et/ou en local.

Installation via npm:

```
npm install -g npm_lazy
```

Vérification:

```
npm_lazy --version
```

Lancement du serveur:

```
npm_lazy --config ./npm_lazy_config.js
```

Configuration de npm (globalement ou localement):

```
npm config set registry http://localhost:8077/
```

Vérification

```
npm config get registry  
http://localhost:8077/
```

Restoration de la configuration d'origine:

```
# linux  
npm config rm registry  
  
# windows  
npm config delete registry
```

Vérification

```
npm config get registry  
https://registry.npmjs.org/
```

Ressources: * https://github.com/mixu/npm_lazy

Bower

Bower: outil d'installation des librairies Web

Permet de télécharger et d'installer des librairies Web.

Alternative: browserify (basé sur requireJS et NPM): * <http://browserify.org> *
<http://requirejs.org>

Installation de Bower

Installation globale via npm (super-admin):

```
npm install -g bower
```

Vérification:

```
bower --version  
1.5.2
```

Initialisation bower du projet

Création du fichier bower.json

```
bower init
```

Configuration

Si le proxy d'entreprise est trop restrictif, on peut demander à git d'utiliser https au lieu de git :

```
git config --global url."https://".insteadOf git://
```

Pour installer un composant:

```
bower install angular --save
```

Installer les dépendances bower (bower.json):

```
bower install
```

```
bower angular#~1.4.5      not-cached git://github.com/...
bower angular#~1.4.5      resolve ...
bower angular#~1.4.5      download .../v1.4.5.tar.gz
bower angular#~1.4.5      extract ...
bower angular#~1.4.5      resolved ...
bower angular#~1.4.5      install angular#1.4.5
```

Resources: * <http://bower.io> * <http://bower.io/docs/api/#install>

Surcharge du fichier par défaut

Gulp injecte automatiquement les fichiers déclarés dans la section “main” du fichier bower.json de la librairie.

Il est possible de surcharger cette propriété via l’attribut “override” du fichier bower.json du projet.

Resources:

- (<https://roots.io/using-bootstrap-with-bower-how-to-override-bower-packages/>)

Private Bower

Proxy/cache à déployer sur le réseau local de l’entreprise.

Installation globale via npm:

```
npm install -g private-bower

private-bower --version
1.1.4
```

Configuration du projet (fichier .bowerrc):

```
json      {      "registry":
"http://bowerproxy:5678",      "timeout": 300000 }
```

Resources:

- <https://github.com/Hacklone/private-bower>

Gulp

Outil d'automatisation des tâches: * concaténation et minification JS/CSS *
injection automatique des librairies * création du fichier source maps * exécution
des tests unitaires * serveur Web, proxy * rechargement automatique en dev *
...

Installation via npm:

```
bash      npm install -g gulp
```

Notez l'option “-g” pour installer l'outil globalement. Requier les droits de
super utilisateur.

Vérification:

```
gulp --version  
CLI version 3.9.0
```

Lancement de la tâche par défaut (ici, démarrage du serveur Web en mode dev):

```
gulp  
[12:57:36] Using gulpfile ~/Documents/dev/formation-angular/2015-09_rabat/tp-git/gulpfile.js  
[12:57:36] Starting 'connect'...  
[12:57:36] Finished 'connect' after 18 ms  
[12:57:36] Starting 'watch'...  
[12:57:36] Finished 'watch' after 11 ms  
[12:57:36] Starting 'default'...  
[12:57:36] Finished 'default' after 4.39 s  
[12:57:36] Server started http://localhost:3000  
[12:57:36] LiveReload started on port 35729
```

Ouvrir l'application: * <http://localhost:3000>

Ressources : * <http://gulpjs.com>

Yeoman

Générateur d'applications

Générateur d'application Web:

- basé sur bower/grunt/gulp
- démarrage instantané du projet
- avec les meilleures pratiques

Les générateurs:

- – de 2000 générateurs
- classés par votes sur github
- Possibilité de créer son propre générateur

Resources:

- [Liste des générateurs](#)

Swiift generator gulp-angular

Génère une application angular + gulp/bower/npm:

- pré-intégration bower/angular des librairies les plus courantes
- best-practices pour l'arborescence et la syntaxe des fichiers
- intégration auto librairies bower
- production: concaténation, minification, uglyfication des css/js
- dev: proxy vers le serveur hébergeant l'api REST
- dev: rechargement à chaud des fichiers dans le navigateur
- compression des images
- transpiler les scss
- génération du source maps
- exécution des tests

Utilisation

```
yo gulp-angular
? Which version of Angular do you want? 1.4.0 (stable)
? Which Angular's modules would you want to have? (ngRoute and ngResource will be ad
? Would you need jQuery or perhaps Zepto? None (Angular will use its own jqLite)
? Would you like to use a REST resource library? None, $http is enough!
? Would you like to use a router ? UI Router, flexible routing with nested views
? Which UI framework do you want? Bootstrap, the most popular HTML, CSS, and JS fram
? How do you want to implements your Bootstrap components? AngularStrap, AngularJS 1
? Which CSS preprocessor do you want? Sass (Node), Node.js binding to libsass, the C
? Which JS preprocessor do you want? None, I like to code in standard JavaScript.
? Which html template engine would you want? None, I like to code in standard HTML.
```

Resources:

- [generator-gulp-angular](#)

SCSS: css preprocessor

- préprocesseur css
- compatible css (courbe d'apprentissage aisée)
- variables, fonctions (mixin)

Resources:

- [sass/scss guide](#)

Formateur

Nicolas TOUBLANC nicolas@toublanc.org +33 6 30 82 27 73 twitter: @toubi-web