Créer le repository sur github

Installer Node.js pour avoir npm

<https://docs.npmjs.com/getting-started/installing-node>

installer gulp

<http://gulpjs.com/>

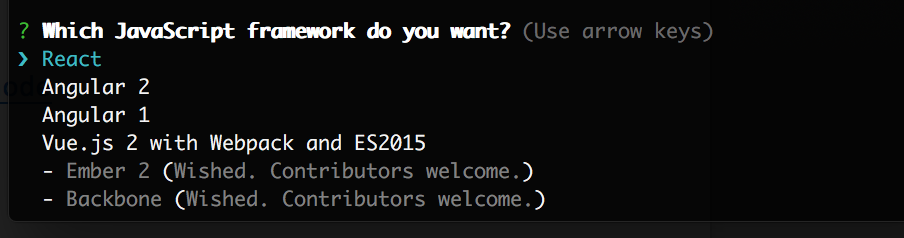
installer fountain-generator

<http://fountainjs.io/doc/usage>

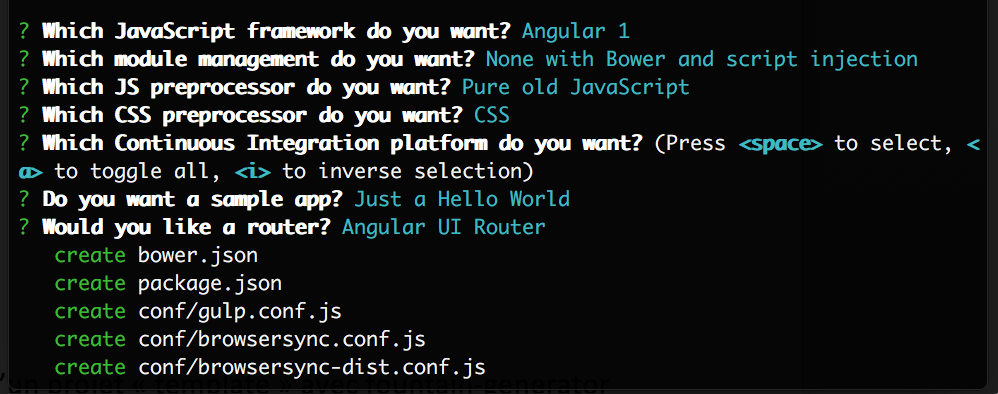
Dans le repository, démarrer la génération d’un projet « template » avec fountain-generator

yo fountain-webapp

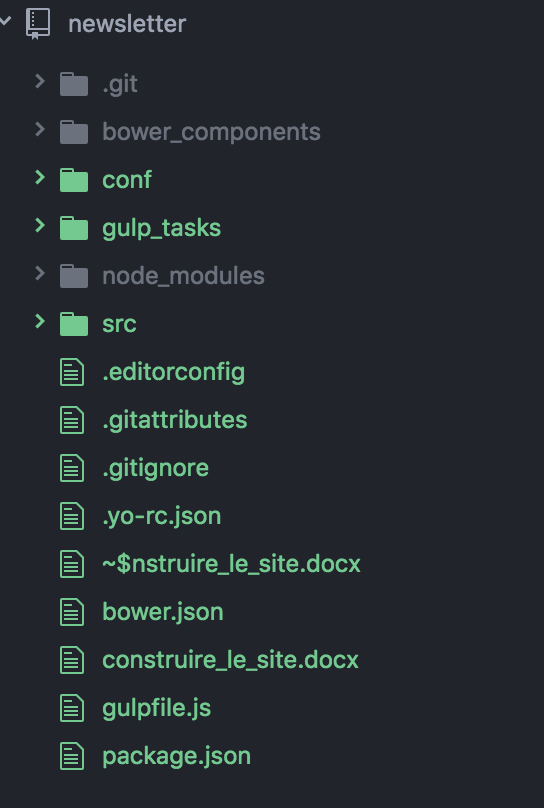
Répondre aux questions posées



Voici les choix que je fais (il s’agit de choix simples car je ne connais pas tous les outils autour de javascript).



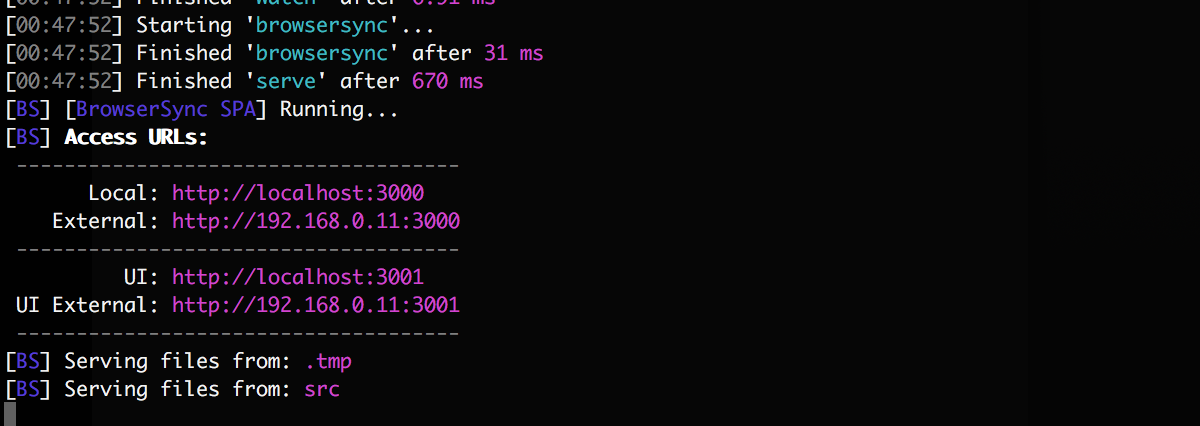
Voici ce que contient maintenant mon repository :



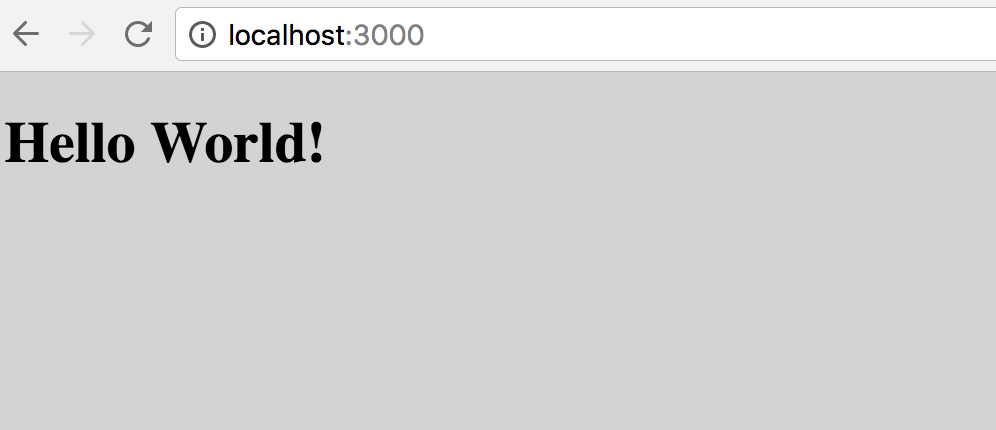
Tester l’application démo

(cf <http://fountainjs.io/doc/usage)>

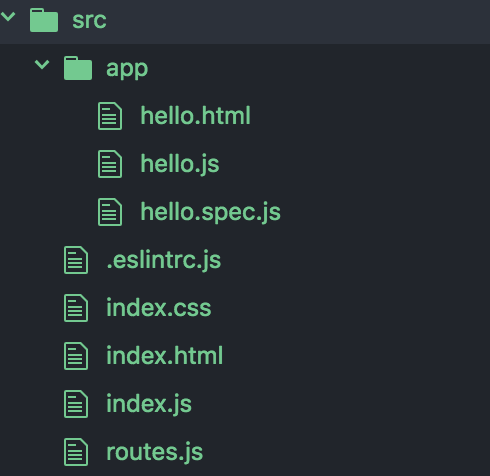
benjaminhabert$ gulp serve



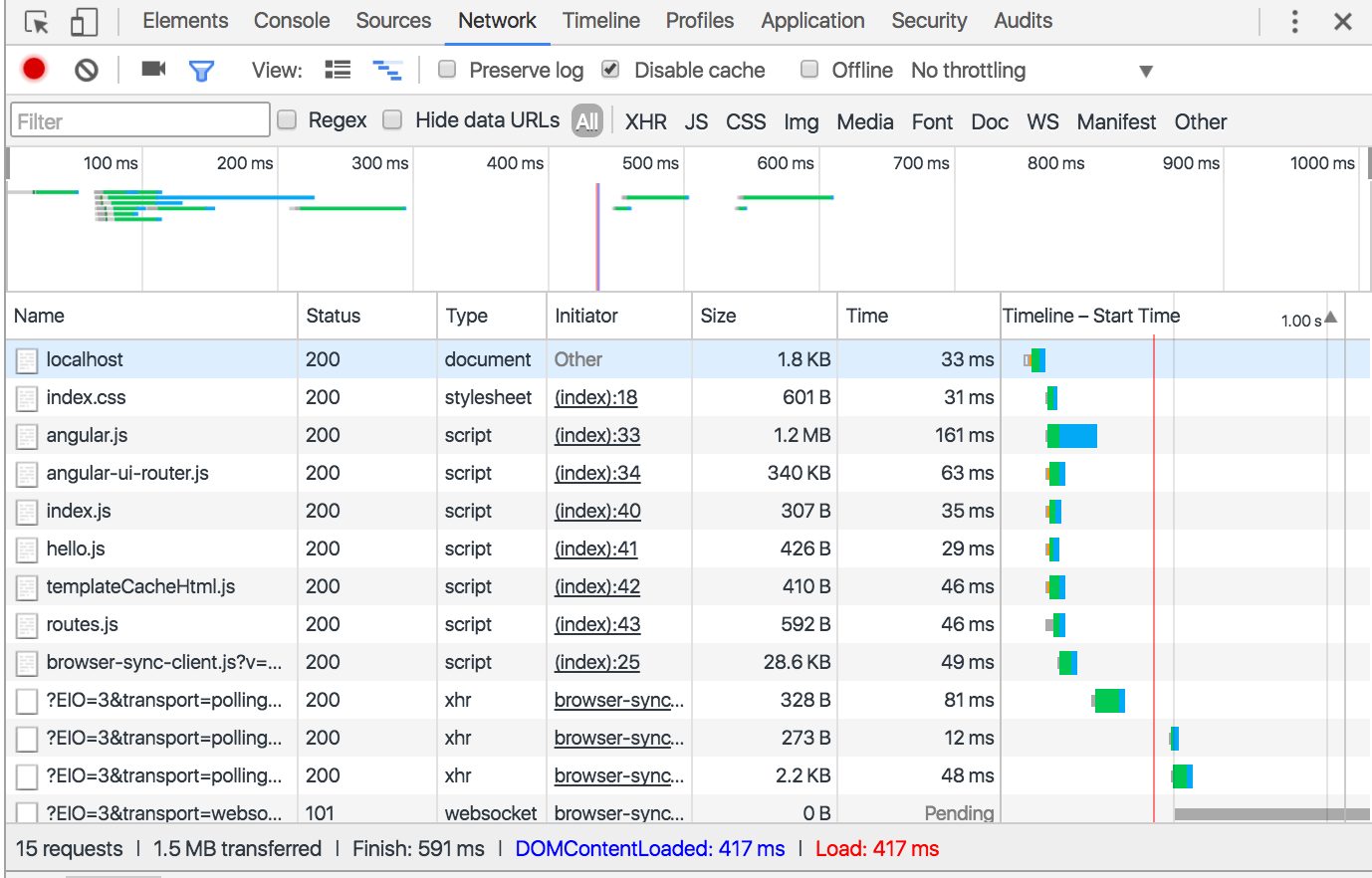
Je vais voir mon navigateur



Ce qui correspond aux fichiers :



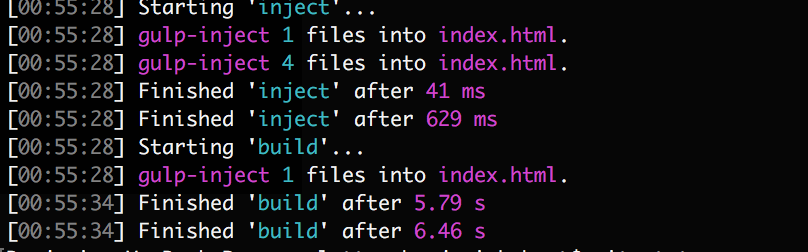
Processus de chargement



Distribuer l’application

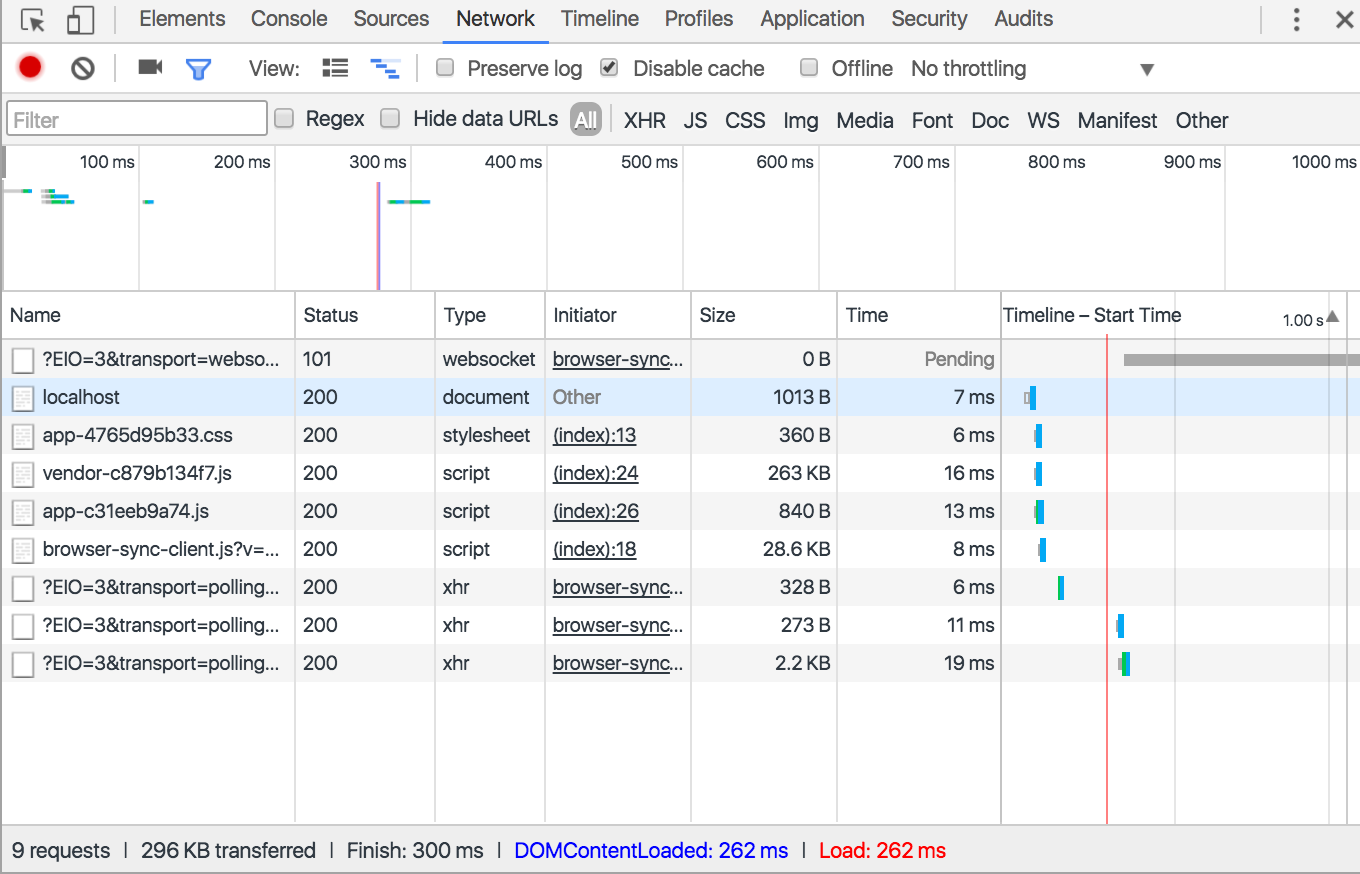
Etape 1 : build

$ gulp build



$ gulp serve:dist

Perfs de chargement :



Etape 2 – publier normalement

$ python -m http.server

$ python –m SimpleHTTPServer

Etape 2 bis - Publier le site sur github

Installer un plugin :

<https://github.com/shinnn/gulp-gh-pages>

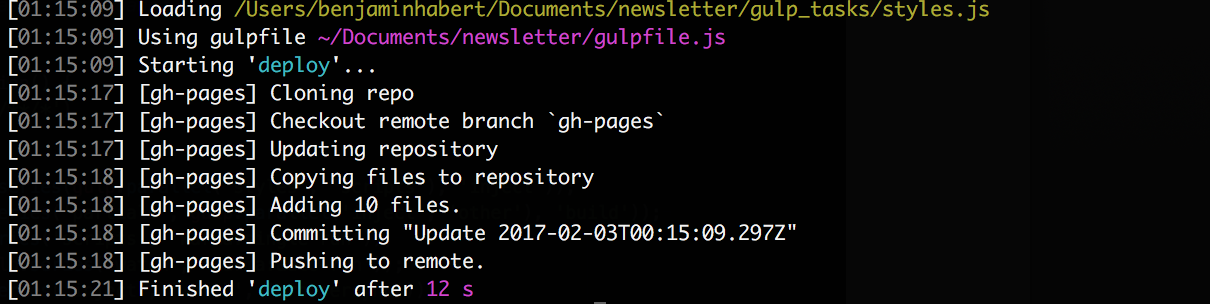
npm install --save-dev gulp-gh-pages

Modifier le fichier gulpfile.js, voici ce que j’ai maintenant

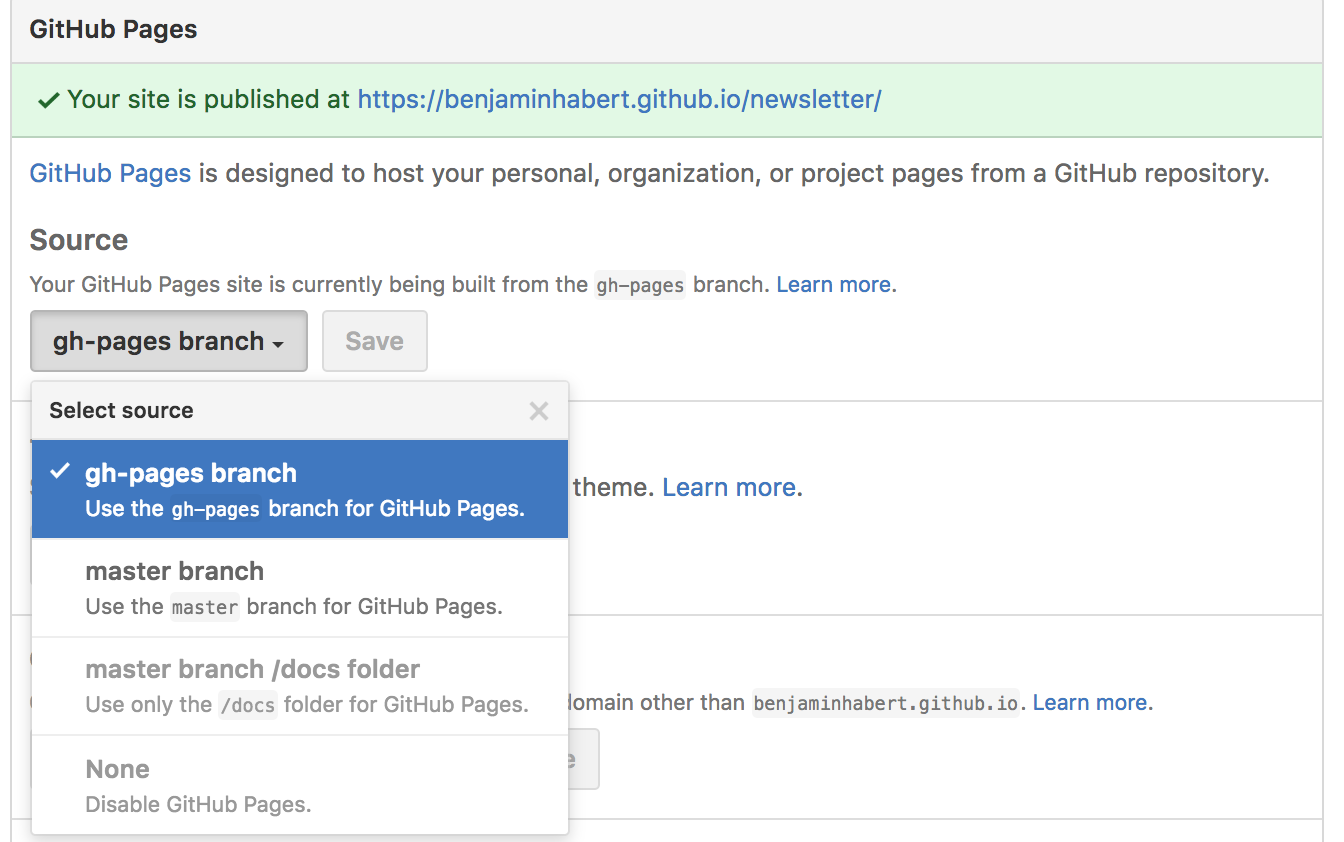
(ligne 2 est nouvelle ainsi que les lignes 14 à 17)



$ gulp deploy

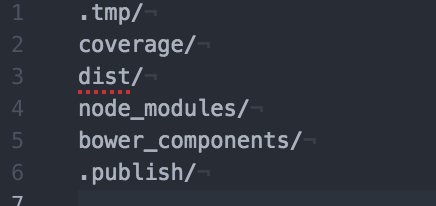


Configurer github-pages :

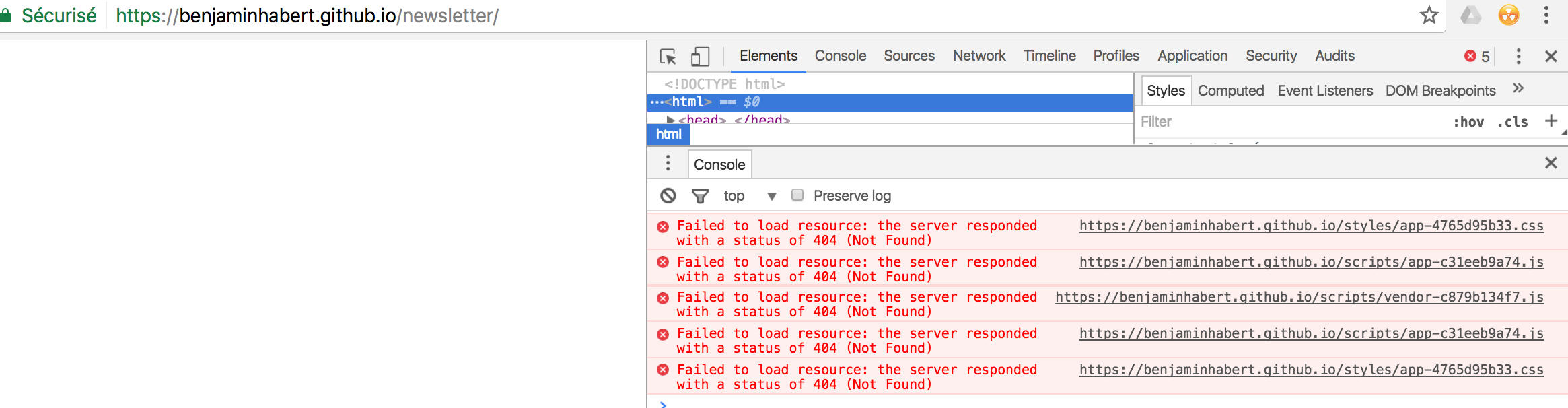


Au passage, je conseille de supprimer le répertoire .publish de git

Le fichier .gitignore :

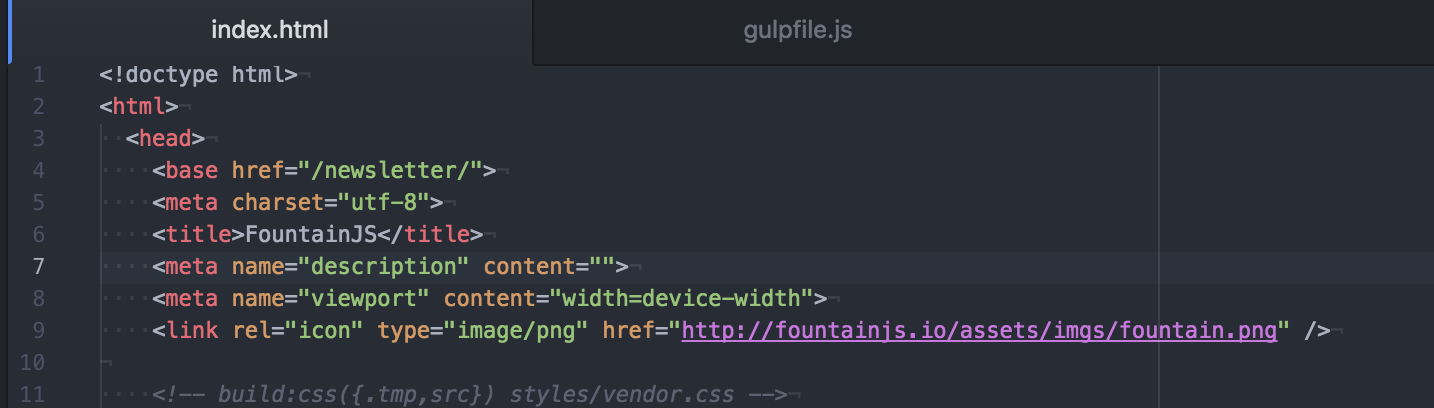


Problème :



Solution :

Modifier l’adresse de « base » :



Note : je suis obligé de laisser « / » quand je fais du développement en local chez moi. Il faut penser à remettre « /newsletter/ » quand je veux gulp deploy

# Premiers elements de code

J’installe des choses pour faire du bootstrap

$ bower install angular-bootstrap --save

<https://github.com/angular-ui/bootstrap>

(la version angular de bootstrap)

Quand on fait bower install -> ça télécharge la librairie dans bower\_components/

Quand on fait --save -> ça ajoute la dépendance dans le fichier bower.json

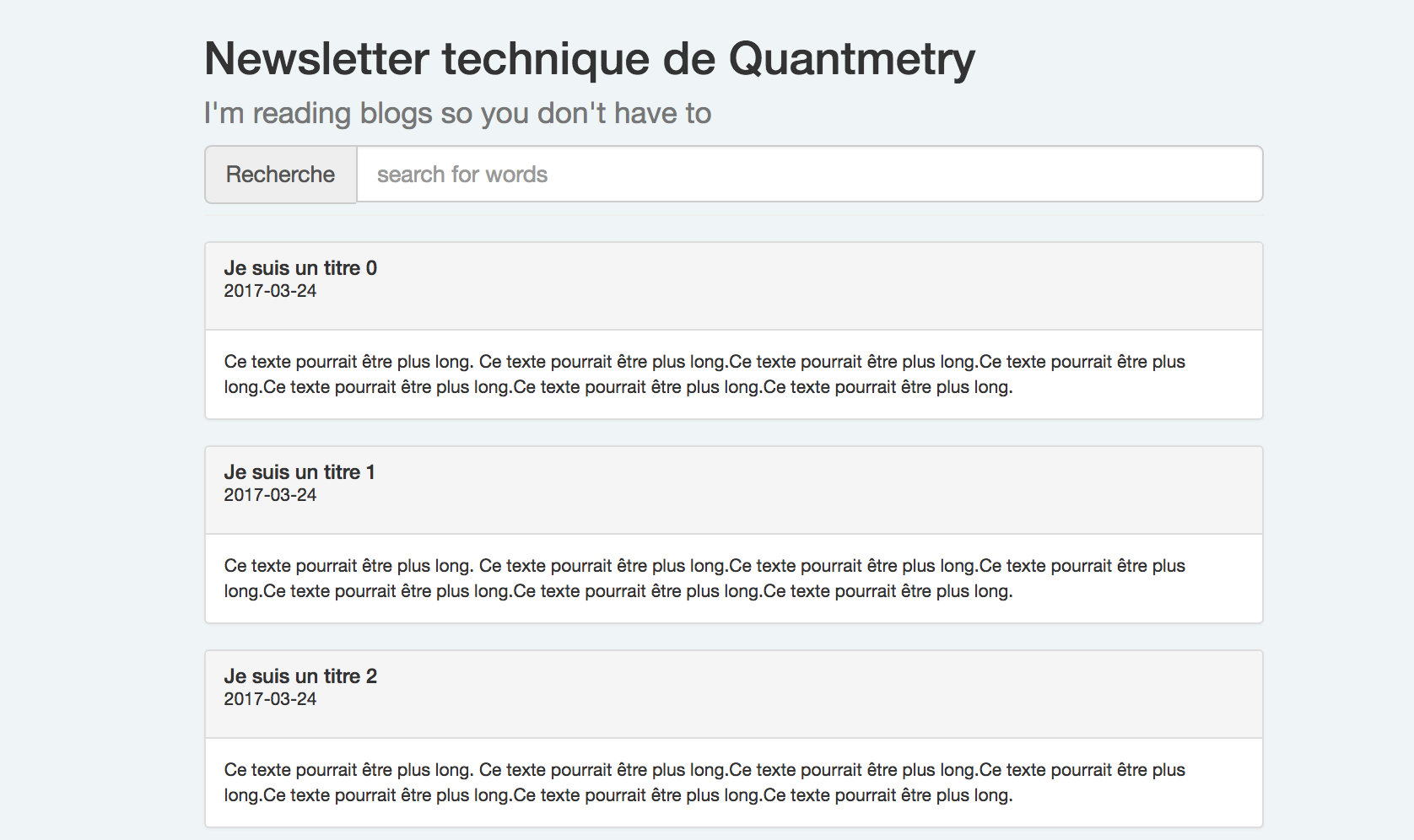
Il faut également installer les dépendances css, j’ai décidé d’installer

$ bower install bootstrap-only-css --save

Maintenant, pour utiliser angular-bootstrap dans l’application :

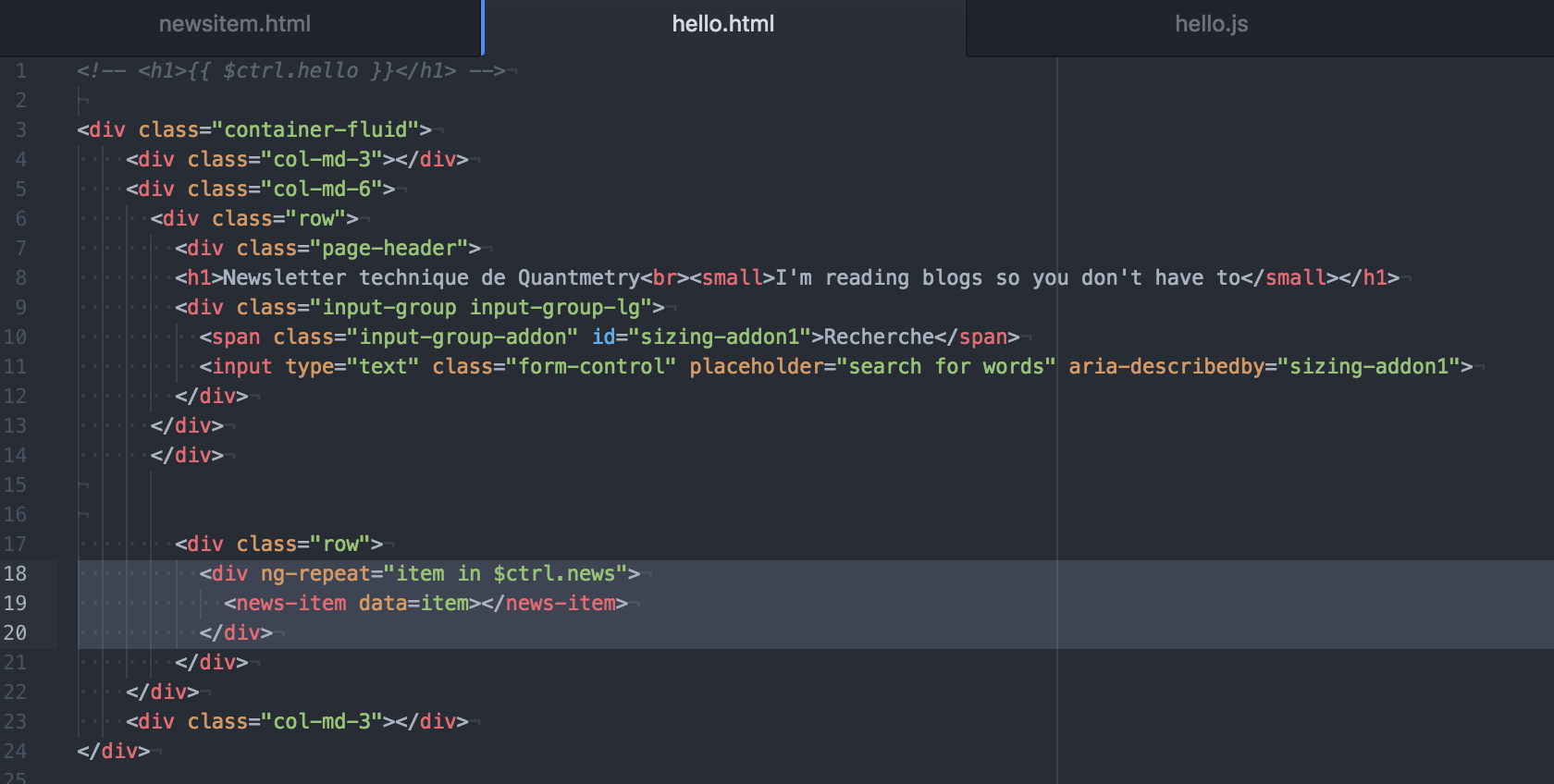
angular.module('myModule', ['ui.bootstrap']);

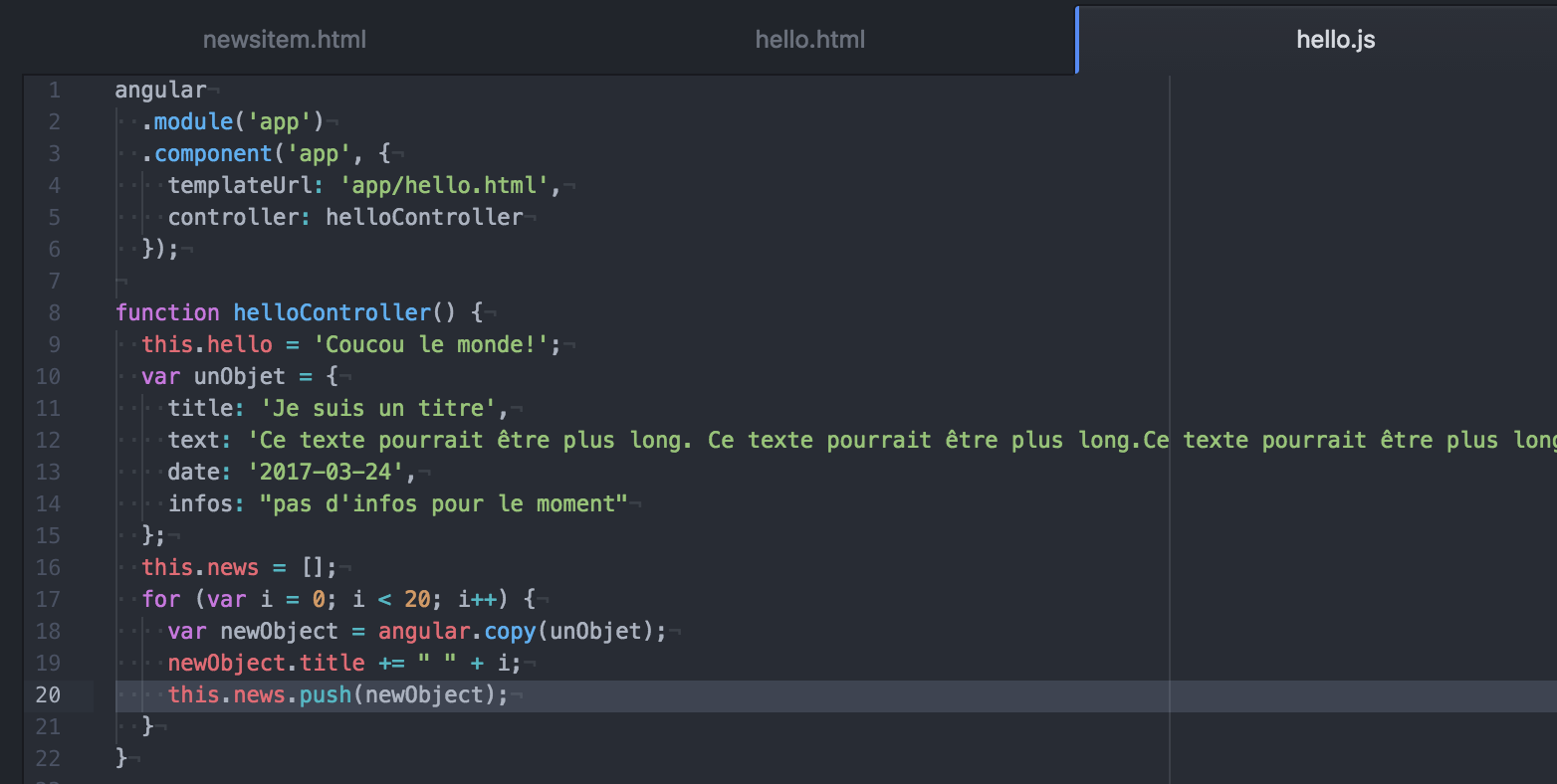
(dans index.js -> il faut ajouter cette dépendance ; c’est ça qui fait l’importation)



Dans la partie « main » de l’application (hello.html et hello.js)

On ajoute un « page header » qui contient une barre de recherche (ne fait rien pour l’instant). Mais surtout on ajoute un ng-repeat sur l’objet « ctrl.news ». Voir dans le code javascript : il s’agit simplement d’un array qui contient 20 éléments. Chaque élément (item dans la boucle) est un élément avec un titre, un texte et une date.

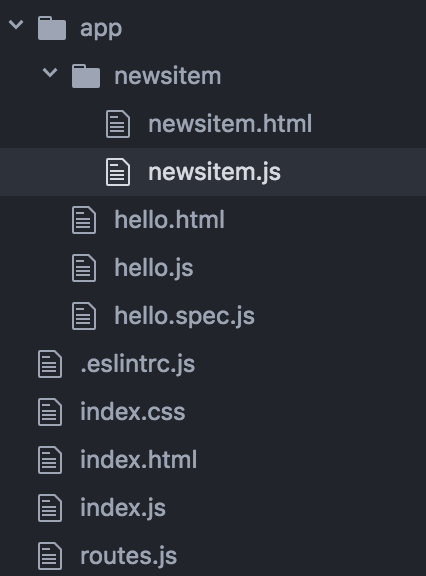


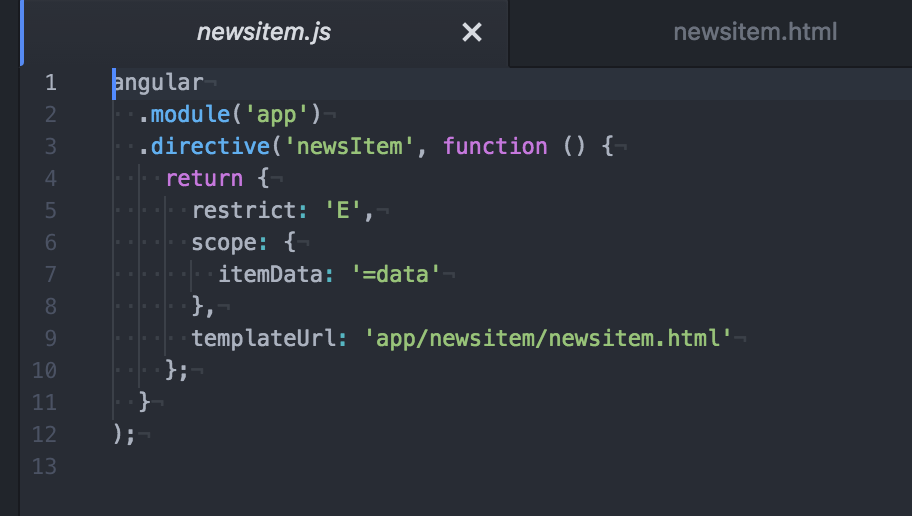


On utilise la directive « news-item » pour représenter chaque élément de news. C’est une directive que nous avons créée nous même.

A noter :

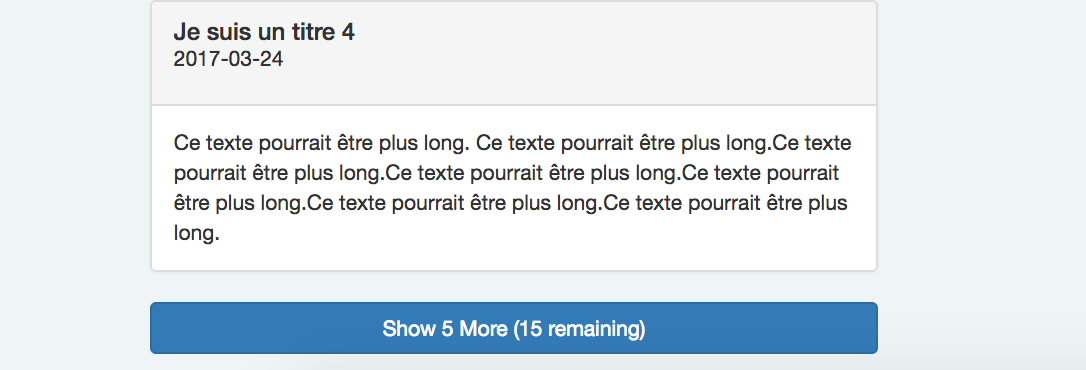
* Dans le scope de la directive, on utilise itemData : ‘=data’ . C’est ça qui nous permet de fournir les données pour cette directive (<news-item data=item></news-item> dans la boucle de hello.html)
* On peut ensuite utiliser itemData dans le html de la directive : on montre le titre, la date et le texte.



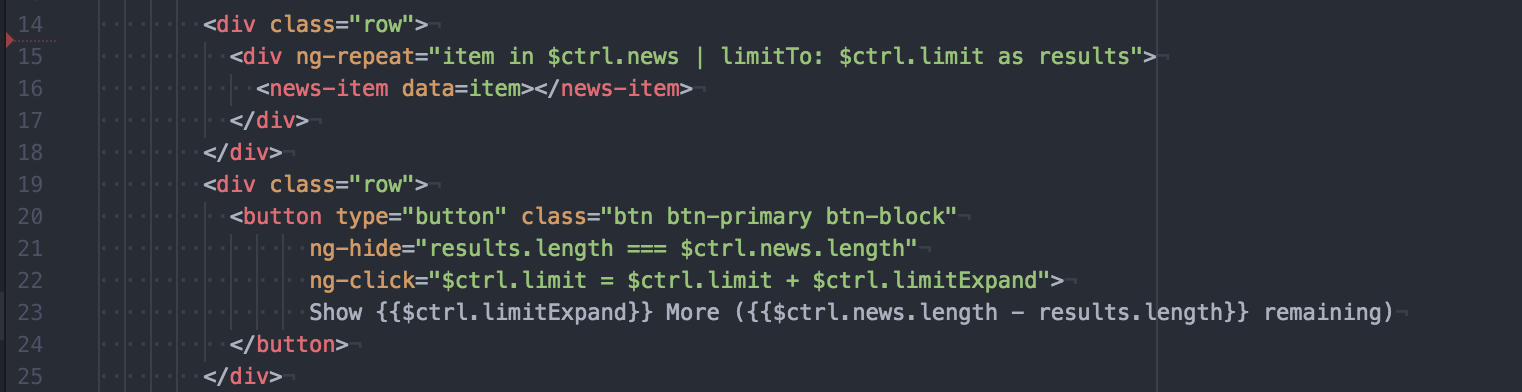




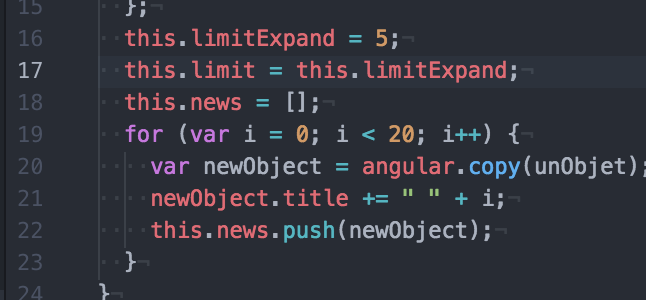
J’ajoute une limite à la liste des news



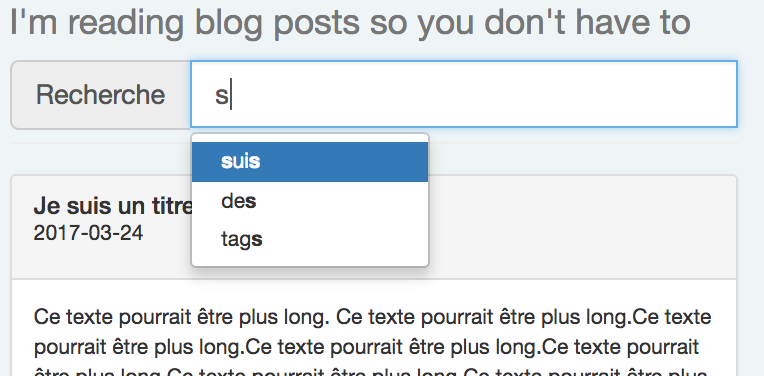
hello.html :



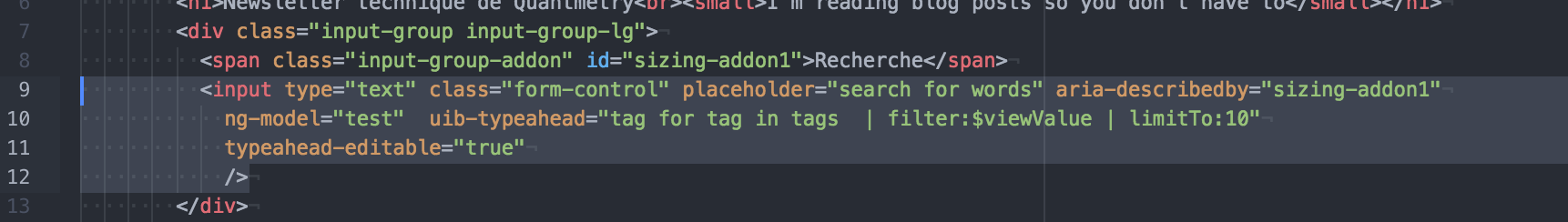
hello.js :



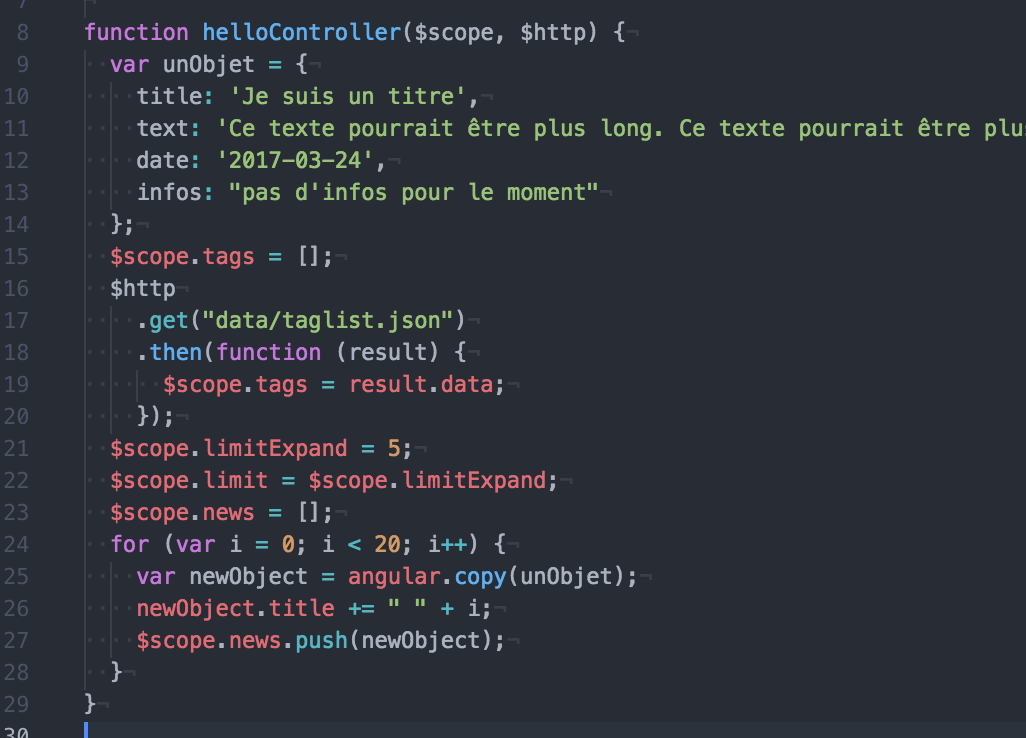
J’ajoute le fait d’avoir une liste de tags dans le champ de recherche pour aider l’utilisateur



J’utilise pour cela l’objet uib-typehead dans hello.html :



Le liste des tags est chargé depuis un fichier JSON  dans hello.js



Note : j’ai modifié le code : j’utilise maintenant $scope au lieu de this

(du coup ça modifie la manière dont j’appelle les variables dans le html : plus besoin de $ctrl)