Contenu

[Documentation  de l’application support de formation  Angular2 1](#_Toc462663061)

[1. how to deploy on Github ? 1](#_Toc462663062)

[2. documentation technique sur l’architecture de l’application utilisable pour le passage de compétence 2](#_Toc462663063)

[1. Main commands available 2](#_Toc462663064)

[2. Application architecture 3](#_Toc462663065)

# Documentation  de l’application support de formation  Angular2

Expliquer le champs script dans package.json (avantage local version vs global npm command) : dans Main commands -> remarks

Npm run jspm vs JSPM install Même chose qu’au dessus

Tester sur un nouveau repertoire Oui pour moi ça fponctionne actuellement

## how to deploy on Github ?

You are working on the branch master and just modified added some slide. Remember to commit/push your work.

First be sure that your project is initialized (follow the README content here: <https://github.com/worldline/TrainingAngular2> )

Then, to deliver, first create the deliverable:

#> npm run dist (ne marche pas sur mon poste -> devrait marcher maintenant)

This will create a dist folder with the full application content, ready for github

Then checkout to the gh-pages and rename the dist folder to gh

#> git checkout gh-pages

#> mv dist gh

Update the gh-pages branch and you’re done

#> git add –A

# git commit –m "update slides"

# git push origin gh-pages

Then check the modifications :

<http://worldline.github.io/TrainingAngular2/gh/#/slide1>

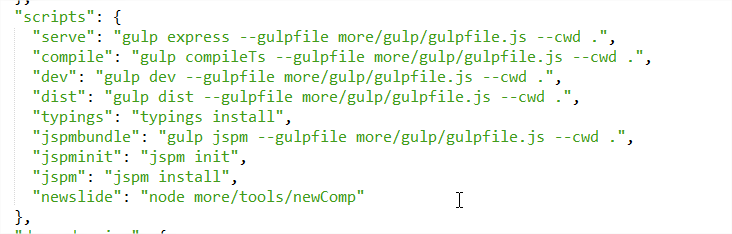
Ajouter commentaires dans Dist-helper.JS -> J’ai commenté tous les fichiers

## documentation technique sur l’architecture de l’application utilisable pour le passage de compétence

### Main commands available

#### Remark

package.json contains the main commands usefull for the project:



Remember that this is interesting to keep a local version of dependencies and avoid the need to get a global version of an executable. For example:

“jspm”: “jspm install” means, use the local version of jspm available in node\_modules to run jspm install.

#### Details

The following shortcuts are available in package.json

* Install jspm dependencies

#> npm run jspm (thanks to script in package.json)

* Compile the application  ne marche pas sur mon poste -> devrait marcher maintenant

#> npm run compile

* Serve the application locally

#> npm run serve

* Serve the application and compile ts each time a modification is detected

#> npm run dev

Ne marche pas sur mon poste -> devrait marcher maintenant

* Create a new slide template (a slide is a component, this will create a new empty component and its template so you can directly start to edit the template instead of create everything manually).

The command takes 2 arguments, the component name and the parent folder in which it has to be created

As an example, the following command will create the **NewComponent** component in the trainingapp/src/components/slides/**parentFolder** folder:

#> npm run newslide – **NewComponent** **parentFolder**

Once the component is created, don’t forget to add your new component into trainingapp/data/slides.json

### Application architecture

#### Bootstrap

* Angular version: rc1
* Entry point of the single app: trainingapp/index.html
* index.html includes:
  + The polyfills needed for rc1 (zone.js, reflect.js, system.js)
  + The system js module loader configuration ( config.js which is also in the trainingapp folder )
  + The load of the entry point module: System.import('src/components/app/app');
  + An <app></app> tag which will be used to load the app component

Manque explication code APP -> ajoutées dans la description du composant App

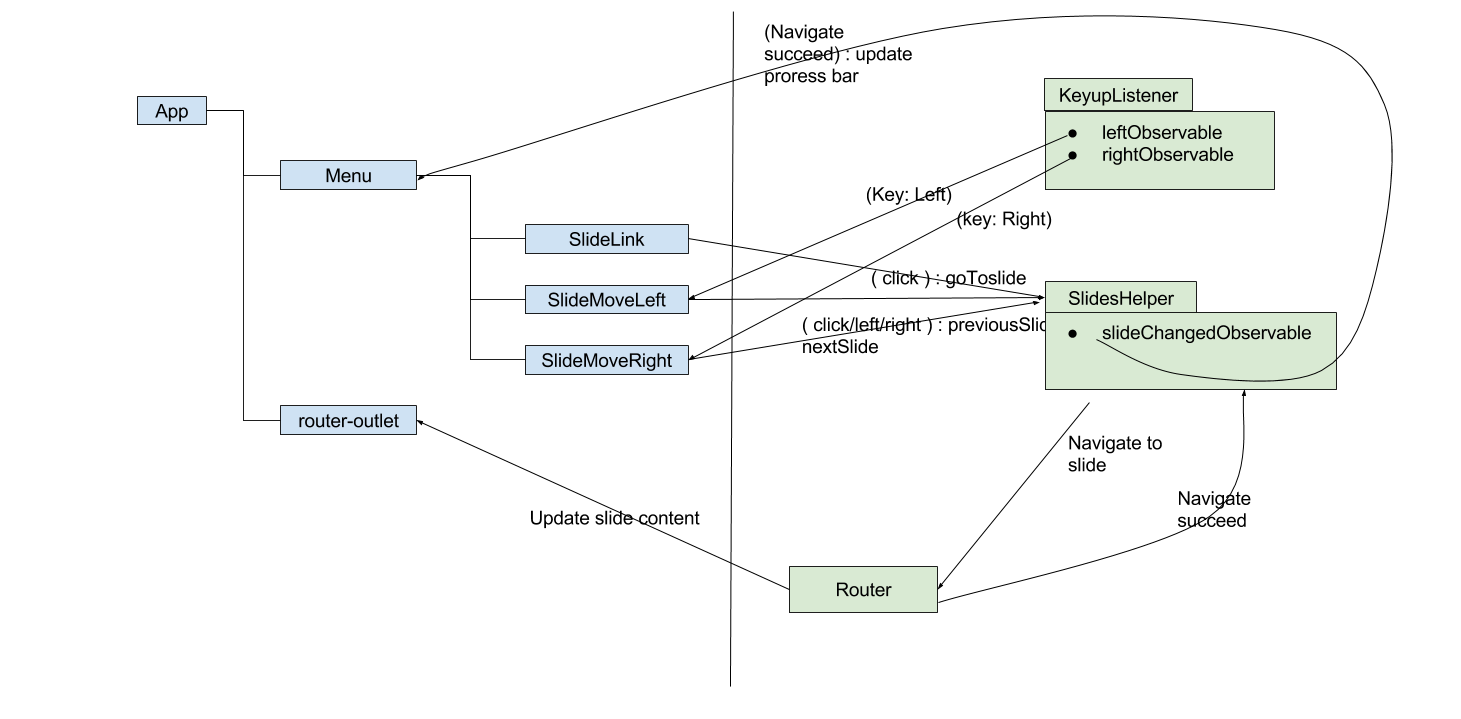
HOST\_SLIDE\_CONTAINER\_CLASS -> Ajouté dans App

GPRETTIFYER -> Ajouté dans App

JQUERY -> Ajouté dans App

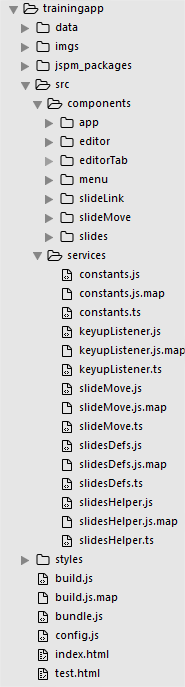
Definition des route. -> Dans slidesHelper

#### Schema

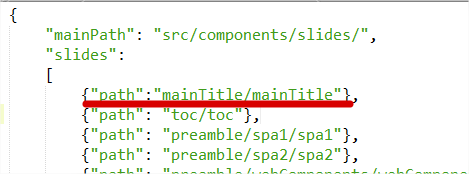


#### Details

* The folder structure is as follow



* + data folder contains slides.json:
    - Each slide is an angular component, let’s see details after
    - Each slide/component which has to be available in your app also has to be declared in slides.json

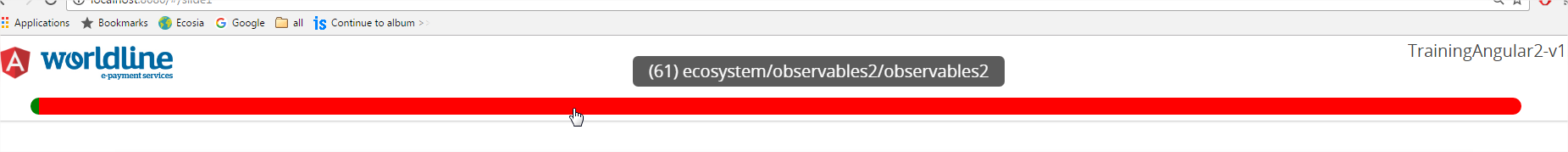


The above example is a part of slides.json which references the component mainTitle

* + imgs: As its name says, contains the web app images
  + jspm\_packages: As its name says, contains the jspm dependencies
  + styles: As its name says contains stylesheets for the web app
  + config.js: contains the jspm config ( configured automatically, shouldn’t be modified manually)
  + You can forget build.js/bundle.js which are created removed by the ”dist” process

Then the src folder contains the code source itself, let’s have a look to this folder into details:

* src/services contains some services:
  + Keyuplistener:
    - this service is instanciated at application bootstrap and listen for left and right key press
    - It then allow any other component/service to subscribe to these streams to listen for key left/right events
    - This will be used by the SlideMoveLeft and SlideMoveRight components to change of slides
  + slideMove.ts:
    - Is a simple abstract class that is implemented by SlideMoveLeft and SlideMoveRight components (so shouldn’t be in the service folder)
  + slidesDefs: Contains the SlideLinkModel class. This is a tool to help which contains informations on a slide (it id: an int which is the page number, the component name associated to the slide, which can be used to create a tooltip for this slide. This is actually not a service but a simple class and could be converted as service or moved to an other folder. This is used by the “menu” component. The menu component is the progress bar (see below) which displays the tooltip provided by SlideLinkModel…



* + slidesHelper:
    - Defines the routes based on the slides.json file
    - Allows to navigate to a slide
    - So it is used by the following components:

Manque un schéma de communication entre les composants

+ des commentaires dans le code source -> fait

* + - * Menu (slideChangedObservable
      * SlideMoveLeft
      * SlideMoveRight

Manque : explication sur Configure slide ZipStatic, AsyncRoute , system.Import -> J’ai mis énormément de commentaires dans le code

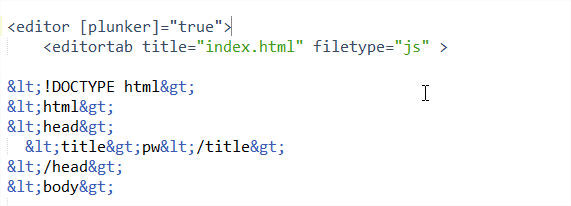
* src/components contains some components:
  + App:
    - It’s the application bootstrap component
    - It’s template contains the menu component and <router-outlet> directive to load the slides (which themselves are components)
    - It also declare some providers, such as
      * HOST\_SLIDE\_CONTAINER\_CLASS : A simple string which corresponds to a css class which is added in the slides container elements
      * GPRETTIFYER : A service containing the google prettify library, used for editor
      * JQUERY : A service containg the jQuery library, used for the editor
  + Menu:
    - Displays the progress bar and can be used to navigate to a slide. The data information (slide number/tooltip) are provided by the SlidesHelper class detailed previously)
    - The progress bar is composed of SlideLink components, each SlideLink corresponds to a link which allows to navigate to a slide
  + SlideMoveLeft / SlideMoveRight: displays buttons used to navigate between previous and next slides and to listen left/right key press with the help of KeyupListener service described previously
  + Editor/ EditorTab:
    - Is based on google prettify: https://github.com/google/code-prettify
    - Allows to create an editor (Editor) composed of tabs (EditorTab) to display js/html code

Manque une vue sur le positionnement des composant dans la page -> le schema est parlant, l’arbre permet de faire le lien avec la vue

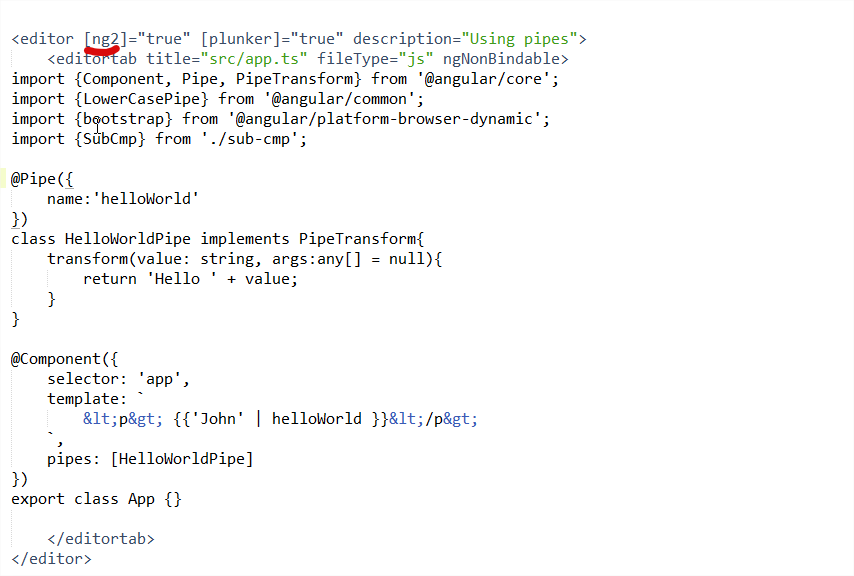


D’ou vient ce code ? -> exemple d’utilisation

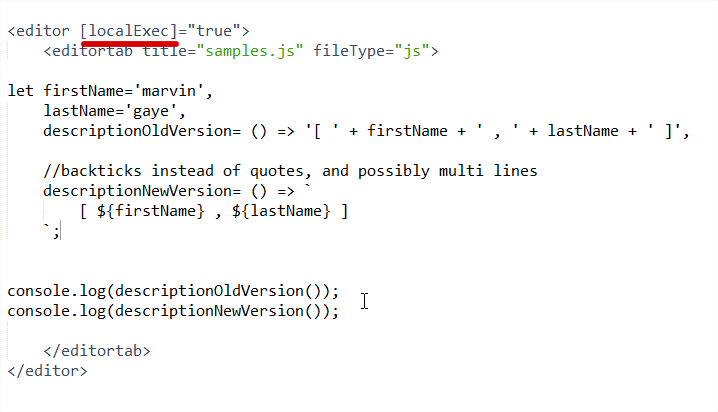
Allows to use plunker D’ou vient ce code ? Manque l’equivalent commentaire dans le JS



* + - Allow to generate an angular app (so you just need to add your specific code for examples, but the ng2 app bases are there )



* + - Allows to execute your JS



* + Slides/\*\*/\*:
    - Contains the slides, each slide is a component, there is nothing important here, each component has its template which corresponds to the slide content. The idea is that if you want to make some ng2 demo, you can do it directly in your slides if you want