

AngularJS - Jour 2

0 - Programme

Au programme:

- Le routing
- Les directives
- TP

1 - Le routing

Routing AngularJS

Documentation officiel : <https://docs.angularjs.org/api/ngRoute>

Manque de fonctionnalité.

Module externe à angular (**angular-route**).

Inclusion via une dépendance module ['ngRoute']

Expose des composants AngularJS:

- provider ([\\$routeProvider](#))
- directive ([ngView](#))
- services ([\\$route](#), [\\$routeParams](#))

Définition d'une route ([link](#)):

- path (pattern url)
- route config (controller, template, resolve, etc.)

[DEMO/day_03/step_01]

Routing UI-router

Projet Github : <https://github.com/angular-ui/ui-router>

Pourquoi utiliser ui-router ?

- Les vues imbriquées
- Les vues multiples nommées
- Liens générique vers un state (et non une url en dur)
- Decorator pour avoir des urls dynamique
- States (Au lieu des routes)

Documentation :

API Reference: <http://angular-ui.github.io/ui-router/site/#/api>

Guide: <https://github.com/angular-ui/ui-router/wiki>

FAQs: <https://github.com/angular-ui/ui-router/wiki/Frequently-Asked-Questions>

Sample Application: <http://angular-ui.github.io/ui-router/sample/#/>

Router BETA angular-new-router

/!\ Version Beta

AngularJS 1.4 apporte un nouveau router, module externe.

<https://github.com/angular/router>

Demo : <https://github.com/angular/router/tree/master/examples/angular-1/hello>

La définition des routes

Le module de router ngNewRouter fournit un service \$router et \$routeParams.

`angular.module('app', ['ngNewRouter'])`

```
.controller('AppController', ['$router', AppController]);
```

```
AppController.$routeConfig([
  {path: '/', component, 'home'},
  {path: '/detail/:id', component, detail}
]);
function AppController ($router) {}
```

// Dans le composant detail
\$routeParams.id = :id de l'url

Les composants

Le nouveau router AngularJS 1.4 utilise des composants (nouveau concept qui permet d'associer un contrôleur et une vue).

La vue

Comment dire à la vue où afficher le composant ?
Avec la directive ng-outlet

Le multi-vue

```
AppController.$routeConfig([
  {path: '/', component, {
    'master': 'home',
    'sidebar': 'navbarHome'
  }
}
]);
```

Les liens entre les vues

ng-link directive : ng-link="detail({id: 5})";

2 - Les directives

Kezako

C'est la base du framework AngularJS.

Simplement : Une directive est une fonction JavaScript qui manipule et ajoute de nouveaux comportements au DOM HTML.

Elles peuvent être simple ou extrêmement compliqué.

Comment AngularJS détecte les directives

Le template HTML peut invoquer une directive AngularJS de quatre manières :

- En tant qu'attribut

- En tant que class CSS

- En tant qu'élément DOM
<my-directive></my-directive>
- En tant que commentaire
<!-- directive: my-directive expression -->

La création

<https://docs.angularjs.org/guide/directive#creating-directives>

Les options

restrict

Elle permet de définir à AngularJS le déclencheur dans le HTML.

'A' : Attribut

``

'E' : Element

`<my-directive></my-directive>`

'C' : Classe

``

'M' : Commentaire

`<!-- directive: my-directive -->`

Template

Permet de définir un template HTML directement dans la directive

template: '`<div class="myclass"></div>`'

TemplateUrl

Permet de définir une url de template HTML à utiliser

templateUrl: 'templates/ng-sparkline-template.html'

Scope

Le scope permet de définir l'héritage.

Nouveau scope

scope: true

Permet de créer un nouveau (qui hérite par prototype de ses parents)

Scope isolé

scope: {}

Permet d'avoir un scope qui n'hérite pas de ses parents

Pour mettre en place une discussion avec l'extérieur on a la possibilité de lui spécifier un comportement par valeur.

Scope local

@ (or @attr)

Bi-directionel

= (or =attr)

Le context parent

& (or &attr)

Controller

Permet de définir un contrôleur

ControllerAs

Permet de nommer le contrôleur sur le scope

bindToController

Permet de partager des données (&, =, @) entre l'extérieur de la directive et de les associer au controller

Les étapes d'initialisation d'une directive

La compilation (\$compile, compile)

Elle s'occupe de la manipulation du DOM avant d'effectuer le rendu.

Manipulation DOM = lent

Le controller

S'occupe de la logique présentation.

Permet d'exposer une API pour cette directive et d'autoriser la communication avec des directives soeur et parente.

[DEMO/day_03/step_02]

[DEMO/day_03/step_03]

3 - TP

Implement le router 2.0

- Ajouter la dépendance *angular-route*
- Utiliser *angular-router* dans notre application
 - Dépendance module
 - Directive de vue
 - Directive de liens
- Configurer les routes suivantes:
 - / ou /home ⇒ Accueil
 - /users ⇒ Page listant les utilisateurs
 - /users/new ⇒ Page avec le formulaire
 - /users/[ID] ⇒ Page avec le détail d'un utilisateur

Ranger le code en composant

/src

-- /components

-- -- /user

/ controller, filter, directives directement reliés à un utilisateur */*

-- -- /home

/ controller, filter, directives directement reliés à l'accueil */*

-- -- /misc

/ controller, filter, directives de type tooling */*

-- index.html

-- app.js

Aller plus loin: ui-router

Utiliser le router ui-router (<https://github.com/angular-ui/ui-router>).

- npm install angular-ui-router
- Utiliser *ui.router* dans notre application