



Développement Web APP2 Angular JS

Thomas Bourdeaud'huy

Mars 2016



ANGULARJS



Quezako ?

- Un framework
 - Pas comme JQUERY...
- Créé en 2010 par Google
- Architecture MVC / MV* / MVW
 - Model, View, Whatever
- Pour des SPA
 - Single Page Applications





Philosophie

- *Le HTML est très bien pensé pour les documents statiques mais il n'est pas du tout adapté pour un usage dynamique tel qu'on le voit dans les applications web.*
- *AngularJS enrichit le HTML avec une grammaire similaire, spécialement conçue pour des RIA.*
- *Le résultat est un code parfaitement expressif (on comprend chaque action effectuée à sa simple lecture), lisible (ça reste du HTML) et facile à écrire.*





Intérêt

- Un grand nombre d'actions effectuées sur le serveur tel que le rendu du moteur de template, la récupération des données, leur (pré) validation et la navigation dans une application, sont désormais déportés côté client.
- Le serveur se limite à traiter, vérifier, valider et envoyer les données aux clients dans un format générique (JSON, XML, etc.)
- Cela permet d'avoir une charge sur les serveurs nettement moins importante et une fluidité de navigation chez le client.





Premiers pas



Installation

- Télécharger
- Charger lib.
- Définir ng-app





Modèle & Vues



Templates

- Déclaration contenant du HTML, des directives et des expressions
 - Les directives définissent comment transformer le template en une vue dynamique
 - Les expressions permettent de mettre en oeuvre le data binding





Expressions

- Affichage de variables à l'aide d'accolades
 - Pas de structures de contrôles
 - ? est permis
- Ex :
`<h1>The sum of 2+3 = {{2 + 3}} ! </h1>`





Directives

- Des balises HTML avec des attributs spécifiques
 - Commencent par ng-*
 - On peut aussi les faire commencer par data-ng-* pour être valide % HTML5
 - <http://www.alsacreations.com/article/lire/1397-html5-attribut-data-dataset.html>
- http://www.w3schools.com/angular/angular_directives.asp
- <https://docs.angularjs.org/api/ng/directive>





Directives

- ng-app
 - Définit une application AngularJS
- ng-model
 - Associe la valeur d'une contrôle HTML (input, select, textarea) aux données de l'application
- ng-bind
 - Associe le contenu d'une balise HTML aux données de l'application





Two-way data binding

- Permet de notifier tous les éléments qui font référence à une variable, de son changement

- Ex :

```
<input type="text" ng-model="yourName">  
<h1>Hello {{yourName}}!</h1>
```

- One-time binding avec ::

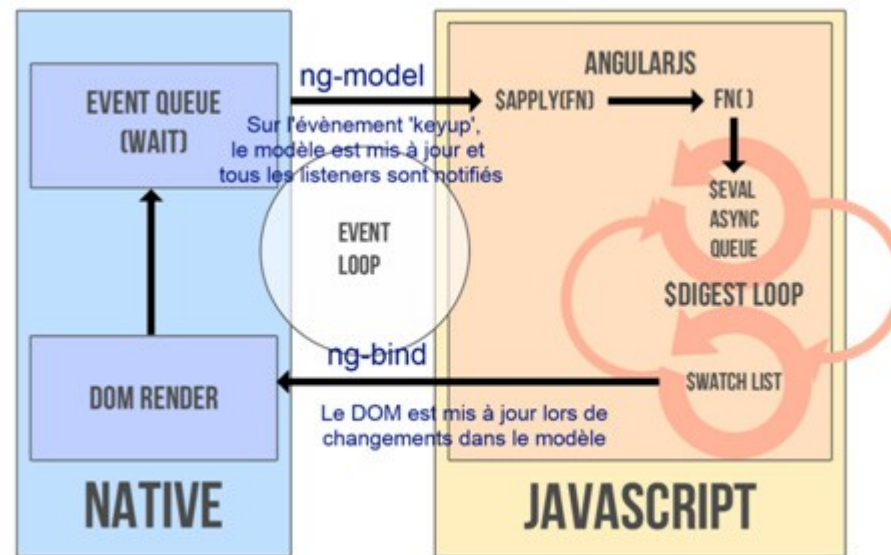
```
<div>{{::message}}</div>
```





Comprendre le Digest Cycle

- <http://www.technologies-ebusiness.com/langages/comprendre-angular-en-refaisant-zero-devoxx-france-2014>





ng-init

- Permet d'initialiser des valeurs pour l'application

```
<div ng-app="" ng-  
init="firstName='John' ">
```

```
<div ng-bind="firstName"></div>
```

```
<input type="text"  
  ng-model="firstName" />
```





Exercice

- Créer des templates permettant de tester data binding & one-time binding





ng-repeat

- Parcours des éléments d'un tableau, duplique l'élément HTML pour chaque case

- Ex :

```
<li ng-repeat="friend in friends">
  {{friend}} </li>
```





ng-repeat

NB: ng-repeat introduit un nouveau scope
\$parent.prop permet d'accéder au scope parent

```
<div ng-app="" ng-init="names=[  
{name:'Jani',country:'Norway'},  
{name:'Hege',country:'Sweden'},  
{name:'Kai',country:'Denmark'}]">
```

```
<ul>  
  <li ng-repeat="x in names">  
    {{ x.name + ', ' + x.country }}  
  </li>  
</ul>  
</div>
```





Exercices

- Créer un menu déroulant à partir des mêmes données
 - Ajouter des identifiants aux données
- Définir dans le modèle une propriété `currentId`
 - Automatiquement mise à jour avec l'identifiant de l'élément sélectionné
- Quel est l'intérêt de `ng-option` ?





ng-class

- Affecte dynamiquement des classes à des éléments en évaluant une expression
- `<element ng-class="expression"></element>`
- Exercice : Sélectionner la classe d'un élément à partir d'un menu déroulant, ou d'une liste de boutons radio





ng-include

```
<body ng-app="">  
<div ng-include="'myFile.htm'"></div>  
</body>
```





ng-show

- `<div ng-app="">`
- `<p ng-show="true">I am visible.</p>`
- `<p ng-show="false">I am not visible.</p>`
- `</div>`

- `ng-disabled, ng-hide`
- Exercice :
 - Suivant la valeur d'une case à cocher, activer un sous-menu





ng-switch

```
<ANY ng-switch="expression">  
  <ANY ng-switch-when="matchValue1">...</ANY>  
  <ANY ng-switch-when="matchValue2">...</ANY>  
  <ANY ng-switch-default>...</ANY>  
</ANY>
```





Filtres

- Utilisés après le symbole “|”, pour les directives ou les expressions
 - Ex : | lowercase ; | uppercase

```
<div ng-app="myApp" ng-controller="personCtrl">  
<p>The name is {{ lastName | uppercase }}</p>  
</div>
```





Filtres natifs

- Sur expressions :
 - currency Format a number to a currency format.
 - date Format a date to a specified format.
 - json Format an object to a JSON string.
 - lowercase Format a string to lower case.
 - uppercase Format a string to upper case.
 - number Format a number to a string.
- Sur directives :
 - orderBy Orders an array by an expression.
 - limitTo Limits an array/string, into a specified number of elements/characters.
 - filter Select a subset of items from an array.





Filter

- filter: critère permet de filtrer les cases d'un tableau lors d'un parcours avec ng-repeat
- |filter : '<car>';
- |filter : <Expression>

- NB : filter, orderBy et limitTo peuvent s'enchaîner





Exercices

- Suggest avec un filtrage
- Filtres pour les tableaux d'objets





Validation de formulaires

- Les champs de formulaire peuvent être dans plusieurs états (propriétés vrai/faux) :
 - \$untouched The field has not been touched yet
 - \$touched The field has been touched
 - \$pristine The field has not been modified yet
 - \$dirty The field has been modified
 - \$invalid The field content is not valid
 - \$valid The field content is valid
- Les formulaires ont les états suivants :
 - \$pristine No fields have been modified yet
 - \$dirty One or more have been modified
 - \$invalid The form content is not valid
 - \$valid The form content is valid
 - \$submitted The form is submitted
- Les classes correspondantes sont positionnées par angular





Exemple

```
<p>Name :  
<input name="myName" ng-model="myName"  
required>  
<span ng-show="myForm.myName.$touched &&  
myForm.myName.$invalid">The name is  
required.</span>  
</p>
```





Style W3schools

- Style W3.css
 - `<link rel="stylesheet" href="`
<http://www.w3schools.com/lib/w3.css>`">`
- Documentation :
 - http://www.w3schools.com/angular/angular_w3css.asp





Logique Métier



Modules

- Un module est un container pour différentes parties d'une application
 - Notamment les contrôleurs
- Un contrôleur appartient toujours à un module
- Il est courant de déplacer les modules dans des fichiers externes

```
<div ng-app="myApp">...</div>
```

```
<script>
```

```
var app = angular.module("myApp", []);
```

```
</script>
```

- Le second paramètre [] permettrait d'indiquer une liste de dépendances à d'autres modules





Contrôleur

- Fonction qui prend généralement en paramètre une référence à \$scope
 - Permet d'avoir accès aux propriétés et méthodes de la vue
 - D'autres dépendances peuvent être passées au contrôleur
- Il peut être défini pour un élément de la vue à l'aide de l'attribut ng-controller
 - Dans ce cas, le contrôleur est invoqué ce qui permet d'initialiser le modèle
 - ng-init est déconseillé...





Module: Contrôleur

```
<div ng-app="myApp" ng-controller="myCtrl">
  {{ firstName + " " + lastName }}
</div>
```

```
<script>
var app = angular.module("myApp", []);
app.controller("myCtrl", function($scope) {
    $scope.firstName = "John";
    $scope.lastName = "Doe";
});
</script>
```





Injection de dépendances

- Lorsque les fonctions (contrôleurs et autres) sont interprétées par l'injecteur de dépendances, celui-ci lit les noms des paramètres, et recherche un service enregistré portant ce nom, qui sera alors fourni comme paramètre lorsque la fonction est appelée
- \$xyz représente un service angular natif





Problématique de la minification

- <http://stackoverflow.com/questions/19238191/understanding-angular-js-controller-parameters>
- Lors de la minification, on renomme les arguments, ce qui risque de casser le mécanisme d'injection de dépendances
- Solution :

```
var myController = ['$scope', '$resource', '$timeout',  
    function($scope, $resource, $timeout) {  
        // this controller uses $scope, $resource and $timeout  
        // the parameters are the dependencies to be injected  
        // by AngularJS dependency injection mechanism  
    }  
];
```





Exercice

- Tester la création d'un contrôleur permettant la minification





Gestionnaires d'événements

```
<div ng-app="myApp" ng-controller="myCtrl">
<button ng-click="myFunction()" ng-dblclick="count=count-1" >Click Me!</button>
<p>{{ count }}</p>
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.count = 0;
    $scope.myFunction = function() {
        $scope.count++;
    }
});
</script>
```





Evenements

- ng-blur
- ng-change
- ng-click
- ng-copy
- ng-cut
- ng-dblclick
- ng-focus
- ng-keydown
- ng-keypress
- ng-keyup
- ng-mousedown
- ng-mouseenter
- ng-mouseleave
- ng-mousemove
- ng-mouseover
- ng-mouseup
- ng-paste



Objet event

```
<div ng-app="myApp" ng-controller="myCtrl">
<h1 ng-mousemove="myFunc($event)">Mouse Over Me!</h1>
<p>Coordinates: {{x + ', ' + y}}</p>
</div>

<script>
var app = angular.module('myApp', []);
app.controller('myCtrl', function($scope) {
    $scope.myFunc = function(myE) {
        $scope.x = myE.clientX;
        $scope.y = myE.clientY;
    }
});
</script>
```





Exercice

- Afficher une trace d'exécution lors du clic sur un bouton
- On peut utiliser le service \$log
 - [https://docs.angularjs.org/api/ng/service/\\$log](https://docs.angularjs.org/api/ng/service/$log)





Custom Directives

- Camel case pour la déclaration, train-case pour l'utilisation
- Restriction possible concernant l'emplacement de la directive
 - E : Element name
 - A : Attribute
 - C : Class
 - M : Comment

```
<div ng-app="myApp" w3-test-directive></div>
<script>
var app = angular.module("myApp", []);
app.directive("w3TestDirective", function() {
    return {
        template : "I was made in a directive constructor!"
    };
});
</script>
```





Custom Filters

```
<ul ng-app="myApp" ng-  
controller="namesCtrl">  
  
<li ng-repeat="x in names">  
    {{x | myFormat}}  
  
</li>  
  
</ul>
```

```
<script>  
var app = angular.module('myApp', []);  
app.filter('myFormat', function() {  
    return function(x) {  
        return x + " (" + x.length + " cars)";  
    };  
});  
app.controller('namesCtrl', function($scope) {  
    $scope.names = [  
        'Jani',  
        'Carl',  
        'Margareth'  
    ];  
});  
</script>
```





Services



Services

- Les services sont des fonctions qui exécutent des tâches communes aux contrôleurs. Ils permettent d'éviter une redondance du code.
 - Ce sont des singletons.
- AngularJS en propose plusieurs par défaut (routing, i18n, ajax, scope, filter, etc.)





Service \$http

```
var app = angular.module('myApp', []);  
app.controller('myCtrl', function($scope, $http) {  
    $http.get("welcome.htm").then(function (response)  
    {  
        $scope.myWelcome = response.data;  
    });  
});
```





Exemple

```
<div ng-app="myApp" ng-controller="customersCtrl">
<table>
  <tr ng-repeat="x in names">
    <td>{{ x.Name }}</td>
    <td>{{ x.Country }}</td>
  </tr>
</table>
</div>

<script>
var app = angular.module('myApp', []);
app.controller('customersCtrl', function($scope, $http) {
  $http.get("http://www.w3schools.com/angular/customers.php")
    .then(function (response) {$scope.names = response.data.records;});
});
</script>
```





Exercices

- Déclencher une requête asynchrone lors du clic sur un bouton
- Déclencher une requête lors de l'initialisation d'un contrôleur, se servir des réponses pour remplir un menu déroulant





\$routeProvider

- Système de gestion de la navigation
- Dans AngularJS, on peut paramétrer le choix du contrôleur et le template choisi pour la vue (valeur en dur ou page externe) pour une route donnée
- Les routes modifieront le contenu de l'élément associé à la directive ng-view du module courant





Exemple

```
app.config(['$routeProvider',
function($routeProvider) {
$routeProvider
.when('/r1', {
    templateUrl: 'templates/r1.html',
    controller: 'r1Controller'
})
.when('/r2', {
    templateUrl: 'templates/r2.html',
    controller: 'r2Controller'
})
.otherwise({
    redirectTo: '/r2'
});
}]);
```





ng-view

- Le lieu où seront chargées les templates
- Une balise ng-view par module
- Animations possibles
- Angular s'occupe de l'historique de navigation !

```
<div ng-view></div>
```

```
<ng-view></ng-view>
```

```
<div class="ng-view"></div>
```





Routes paramétrées





Cookies





Annexes



Favoris

- <https://builtwith.angularjs.org/>
- <https://www.madewithangular.com/#/>
- <https://angularjs.org/>
- <https://www.npmjs.com/>
-





Bibliographie

- Cours de Programmation Web, Présentation du framework AngularJS, Yacine Rezgui – Cédric Ferretti

