

AngularJS

le super-héros en Javaslip rouge



AngularJS, l'histoire



Au début, rien.

1989 HTML - 1995 JS - 2005 AJAX ...

Puis, AngularJS.

“AngularJS c'est la puissance du Flex sans Flex... [#devoxxFR](#)”

#Javascript [#databinding](#) [#multidirectionnel](#)

[#tests](#) **#Google** [#Batarang](#)

Les solutions existantes



Server Side

(CGI, Perl, PHP, Ruby, Django ...)

Structure les pages coté serveur (dumb client)

Abstraction

(Ext js, Sproutcore, Cappuccino ...)

Cache javascript et fait abstraction des different navigateurs.

Manipulation du DOM

(jQuery, Closure ...)

Standardise l'accès au DOM et aux fonctions JS.

Le browser HTML5



ANGULAR

Repond au besoin d'afficher non pas des documents statiques mais des applications HTML5

AngularJS, le concept



Modèle

structure de données, souvent en JSON

Vue

la représentation des données, en HTML tout simple.

Contrôleur

super-héro qui lie données aux vues, en Javascript

AngularJS, sa mission



Découpler

découpler le DOM de la logique applicative

découpler la logique serveur de la logique client

Tester

tester une app HTML5 est infernal. AngularJS permet les tests unitaires.

AngularJS, la logique



Injector

au chargement du DOM un Injector est instancié par ng-app.

Compiler

le service Compile parcourt le DOM à la recherche des directives ng-*

Linker

les directives ng identifiées sont attachées à l'objet \$scope

Demarrer une application



Declarer l'application:

```
<body ng-app>
```

ou bien

```
<div ng-app="AngularDemo"></div>
```

Le controleur

```
<div ng-controller="GreetingsCtrl"></div>
```

...

```
function GreetingsCtrl($scope){};
```

Initialiser une valeur:

```
<p ng-init="greeting = 'World'">Hello: {{greeting}}</p>
```

Ou bien:

```
<!-- MVC style -->
```

```
<p>Hello: {{greeting}}</p>
```

...

```
function GreetingsCtrl($scope){  
    $scope.greeting = "World";  
};
```


Template



Template once, display many:

ng-repeat

Syntaxe proche du *for each in...*

```
<p>Hello: <span ng-repeat="greeting in greetings">{{greeting}}</span></p>
```

...

```
function GreetingsCtrl($scope){  
    $scope.greeting = ["World"];  
};
```

DEMO

GREETINGS

Avec ou sans filtres?



Filtrage délégué:

ng-repeat + filter:*filter source* = filtrage automatique

Filtres customisés:

filters.js

module custom de filtres

Là aussi en quelques lignes de code.

Double rainbow Data-binding



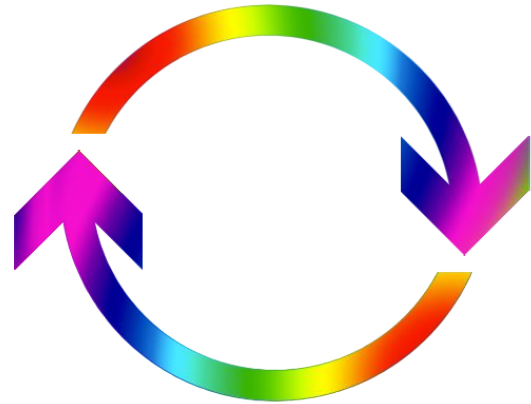
Modèle est modifié:

la vue est mise à jour

La vue est modifiée:

le modèle est mise à jour!

En quelques lignes de code.



Double rainbow Data-binding



```
<div ng-app="AngularDemo">
  <div ng-controller="SuperControleur">
    <input type="text" ng-model="leModele" placeholder="
une vue, un champ texte" />
    <p>Modèle: {{ leModele }}</p>
  </div>
</div>
```

```
var angularDemo = angular.module('AngularDemo', []);
angularDemo.controller(SuperControleur, ['$scope', function
($scope) {
  $scope.leModele = '';
}]);
```

XHR + JSON = epic win!



\$http

service du framework

wrapper des basses besognes XMLHttpRequest

.success ou .error

deux callbacks qui retournent un HttpPromise

Data automatiquement parsé

notre code se concentre sur le binding

function-O-scope



`$scope.zba = function()`

définir une fonction propre à un scope donné

Directives ng pour l'appeler

ng-click, ng-paste, ng-focus, ng-keydown,...

routage



Un design “single page app”

un design désormais classique en HTML5

ng-route et \$routeProvider

configurer le routage notre app

associer à chaque route un contrôleur dédié

Animations



ng-animate

animation par défaut sur directive ng-repeat, ng-view, ng-if,...

formalisme CSS à respecter

un module à inclure, 0 ligne de JS

CONCLUSION



- Leger (80k)
- MVC bien fait
- Moins de code
- Pas de manipulation du DOM
- Modele de données POJO (plain old javascript objects)
- Bonne communauté

CONCLUSION



- Leger (80k)
- MVC bien fait
- Moins de code
- Pas de manipulation du DOM
- Modele de données POJO (plain old javascript objects)
- Bonne communauté