



# HTML ENHANCED FOR WEB APPS!

Présenté par Olivier Combe / [@OCombe](#) / [OCombe](#)

# QU'EST CE QU'ANGULARJS ?

## UN FRAMEWORK JAVASCRIPT PARMY D'AUTRES:

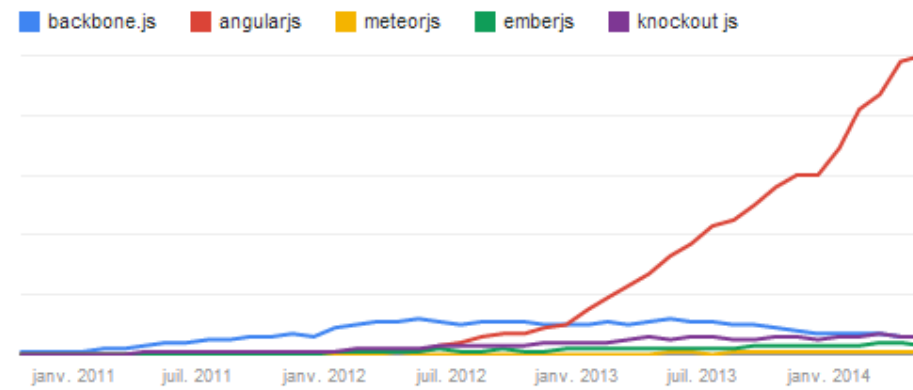
- BackboneJS
- AngularJS
- EmberJS
- MeteorJS
- Knockout

- Développé par Google
- 100% javascript et html (pas de précompilation)
- ~MVC Front-end
- IE8+ jusqu'à 1.2.x, IE9+ à partir de 1.3

## **POURQUOI CHOISIR ANGULARJS ?**

- Permet d'étendre le HTML (directives)
- Binding bi-directionnel
- Inclut jquery lite
- Testable (unitaire & e2e)
- Injection de dépendances
- Très complet
- Très fun (en toute objectivité)
- Très grosse communauté (> 800 000 developpeurs)

Évolution de l'intérêt pour cette recherche. Recherche sur le Web. Dans tous les pays, oct. 2010 - mai 2014.



Google™

Afficher le rapport complet dans  
[Tendances des recherches](#)

## DES INCONVÉNIENTS ?

- Une documentation inégale
- Pas très copain avec le SEO (ça change!)
- Une courbe d'apprentissage chaotique



My Feelings About AngularJS Over Time



# COMMENT ÇA MARCHE ?

```
<!DOCTYPE html>
<html ng-app>
  <head>
    <title>Mon application</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.16/a
  </head>
  <body>
    <input ng-model="name" type="text" placeholder="Ton nom">
    <p>Hello {{ name }}</p>
  </body>
</html>
```

Ton nom

Bonjour

# C'EST BIEN, MAIS EN VRAI, IL EST OÙ LE MVC ?

- On fait un module :
  - Isolé (équivalent d'un namespace)
  - Partageable
  - L'application est elle même un module

```
<html ng-app="MyApp">  
  ...  
</html>
```

```
var myApp = angular.module('MyApp', []);
```

Et on ajoute un controleur :

```
<html ng-app="MyApp">
  <div ng-controller="MyCtrl">
    Secondes: {{ seconds }}
  </div>
</html>
```

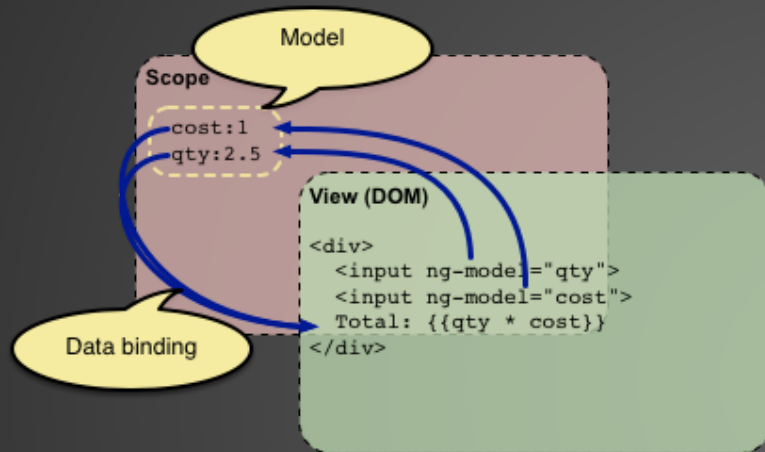
```
myApp.controller('MyCtrl', function($scope, $interval) {
  $scope.seconds = 0;

  $interval(function() {
    $scope.seconds++;
  }, 1000);
});
```

Secondes: 13

# QU'EST CE QUE \$SCOPE ?

- l'équivalent des modèles de l'application et utilisant un héritage répliquant la structure du DOM (\$rootScope)



- un moyen d'échanger des données entre vue et contrôleur
- le contexte d'exécution des expressions (\$apply)
- un moyen d'observer des expressions (\$watch, dirty checking)
- un moyen de propager/recevoir des événements (de façon hiérarchique, \$emit / \$broadcast / \$on)

# LES EXPRESSIONS

- Toute propriété ou méthode appelée dans une expression fait référence au `$scope` (et à ses parents)
- `{{ seconds }}` équivaut à `$scope.seconds`
- On peut utiliser
  - les opérateurs (`=`, `==`, `===`, `!=`, `<`, `<=`, `+`, `-`, `*`, `%`, `&&` ...)
  - les mots clés (`true`, `false`, `null`, `undefined`)
  - les expressions ternaires (`value ? res1 : res2`)
  - les filtres (`data | filterA`)

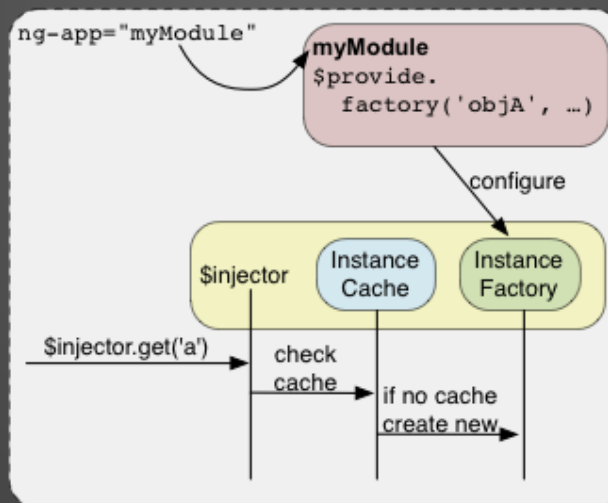
```
{{ today | date:'dd/mm/yyyy' }}
```

# INJECTION DE DÉPENDANCES (DI)

- Un design pattern permettant aux composants d'accéder à leur dépendances

```
var myApp = angular.module('MyApp', ['ui.router']);
```

```
myApp.controller('MyCtrl', function($scope, $interval) {  
    ...  
});
```





# ATTENTION À LA MINIFICATION...

```
myApp.controller('MyCtrl', function($scope, $interval) {  
    ...  
});
```

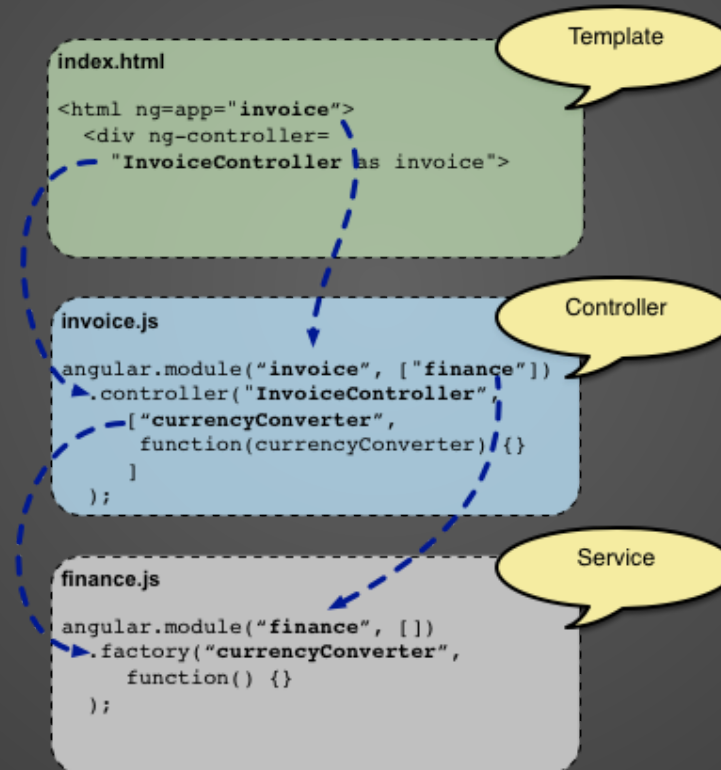
```
a.controller('MyCtrl', function(b, c) {  
    ...  
});
```

```
myApp.controller('MyCtrl',  
    ['$scope' , '$interval', function($scope, $interval) {  
        ...  
    }]  
);
```

```
a.controller('MyCtrl', ['$scope' , '$interval', function(b, c) {  
    ...  
}]);
```

# LES SERVICES

- Gère tout ce qui est indépendant des vues
- Est défini une fois puis injecté à la demande



- Permet de partager des données entre contrôleurs

```
var app = angular.module('myServiceModule', [])
```

```
app.controller('MyController',  
  ['$scope', 'notify', function ($scope, notify) {  
    $scope.callNotify = function(msg) {  
      notify(msg);  
    };  
  }]  
)
```

```
app.factory('notify', ['$window', function(win) {  
  var msgs = [];  
  return function(msg) {  
    msgs.push(msg);  
    if (msgs.length == 3) {  
      win.alert(msgs.join("\n"));  
      msgs = [];  
    }  
  };  
}]);
```

# LES DIRECTIVES

- Sont appelées via un marqueur dans le DOM (attribut, élément, commentaire ou classe CSS)
- Sont compilées lors de leur insertion dans le DOM
- Peuvent attacher un comportement spécifique au DOM

```
<button ng-click="doSomething()"></button>
```

- Peuvent modifier le DOM

```
<modal></modal>
```

```
<div class="modal"></div>
```

- Peuvent surcharger des éléments html existants
- Peuvent accéder au \$scope et le manipuler ou en créer un nouveau isolé
- Peuvent accéder aux services





# ANGULARJS EST

- Riche
- Novateur
- Puissant
- Simple à prendre en main, difficile à maîtriser

## QUELQUES LIENS UTILES

- [Doc AngularJS](#)
- [The Hitchhiker's Guide to the Directive](#)
- [Collection de liens utiles](#)
- [AngularJS & SEO](#)
- [AngularJS modules](#)
- [Angular Seed](#)



# QUELQUES MODULES UTILES

- UI Router
- UI Bootstrap
- Angular Translate
- Angular LocalForage