



AngularJS



- Un framework pour structurer les applications web
- HTML est un langage déclaratif permettant de créer des pages statiques, AngularJS est également déclaratif mais permet de créer des applications dynamiques
- Adapté pour des applications de type CRUD
- Créé en 2009 par Miško Hevery et Adam Abronsw puis repris par Google
- Sous licence MIT

AngularJS - Installation



- Téléchargement

<https://code.angularjs.org/>

- CDN

<https://ajax.googleapis.com/ajax/libs/angularjs/1.6.5/angular.min.js>

- Bower

```
bower install angular
```

- npm / Yarn

```
npm install angular
```



- Live Reload

```
npm install -g live-server  
live-server
```

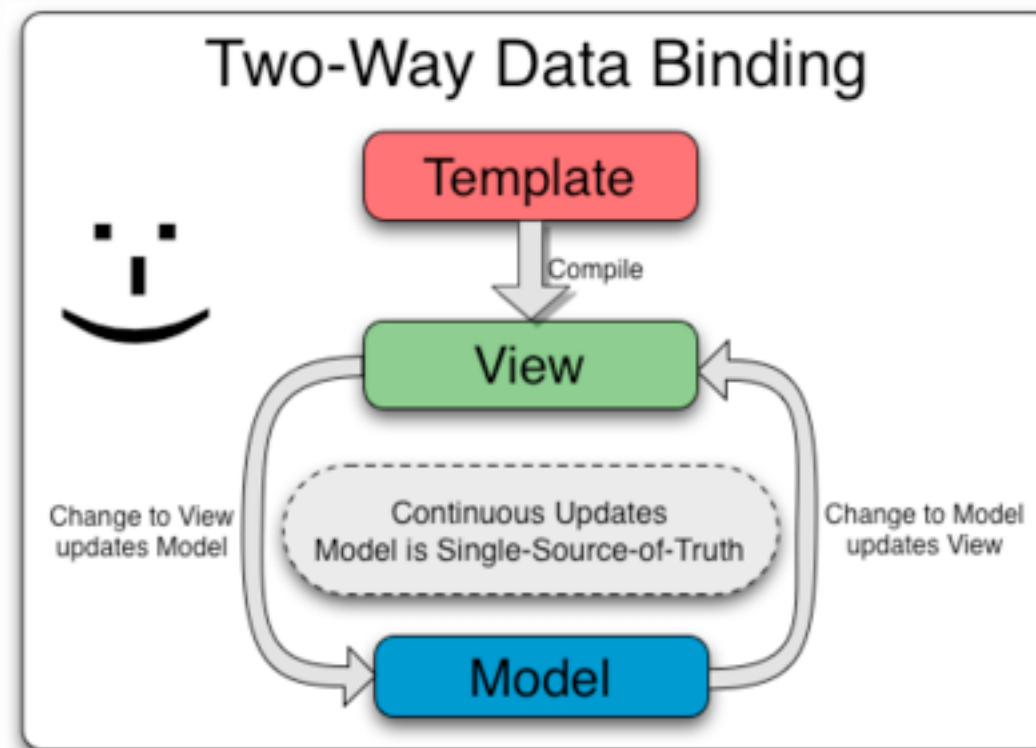
- webpack-dev-server

- SystemJS

- Apache, nginx...



- ▶ Model View ViewModel (MVVM)
Design Pattern introduit par Microsoft en 2005 dans Windows Presentation Foundation et Silverlight
- ▶ Facilite le développement d'interface graphique
- ▶ Two-way data binding
Le model peut mettre à jour la vue, la vue peut mettre à jour le model.



AngularJS - Hello, World !



- Directives

Balises ou attributs HTML qui compilent en JS (ex : ng-app et ng-model)

- ng-app

Directive qui déclare la racine de l'application (en général <html> ou <body>)

- ng-model

Directive qui lie le contenu d'une balise input, select ou textarea à une variable

- {{yourName}}

Lie cette partie de la vue à la variable youName (peut être une expression)

```
<!DOCTYPE html>
<html ng-app>
<head>
  <script src="bower_components/angular/angular.js"></script>
</head>
<body>
<div>
  <label>Name:</label>
  <input type="text" ng-model="yourName" placeholder="Enter a name here">
  <h1>Hello {{yourName}}!</h1>
</div>
</body>
</html>
```



▸ ngApp

La directive ngApp désigne la balise racine de l'application en général <body> ou <html>. Elle permet également de spécifier un module qui deviendra le module racine de l'application.

```
<html ng-app="appFilmotheque">
```

▸ ngController

La directive ngController permet d'associer le contrôleur à la vue (peut également se faire via des routes)

```
<body ng-controller="FilmController">
```

▸ ngModel

La directive ngModel lie une balise input, select ou textarea à une propriété du scope.

```
<input type="text" ng-model="film.titre">
```



▸ ngIf

Permet de créer l'élément du DOM selon une condition

```
<div class="col-sm-9" ng-if="showFilm">
  <h2>{{showFilm.titre}}</h2>
  ...
</div>
```

▸ ngRepeat

Permet de répéter l'élément du DOM pour chaque élément d'un tableau

```
<li role="presentation" ng-repeat="film in films">
  <a href="#" ng-click="show(film)">{{film.titre}}</a>
</li>
```

▸ ngClick, ngChange, ngSubmit, ng...

Permet de lier à un événement du DOM

```
<li><a href="#" ng-click="showFilm = null">Ajouter</a></li>
```


AngularJS - Directives Personnalisées



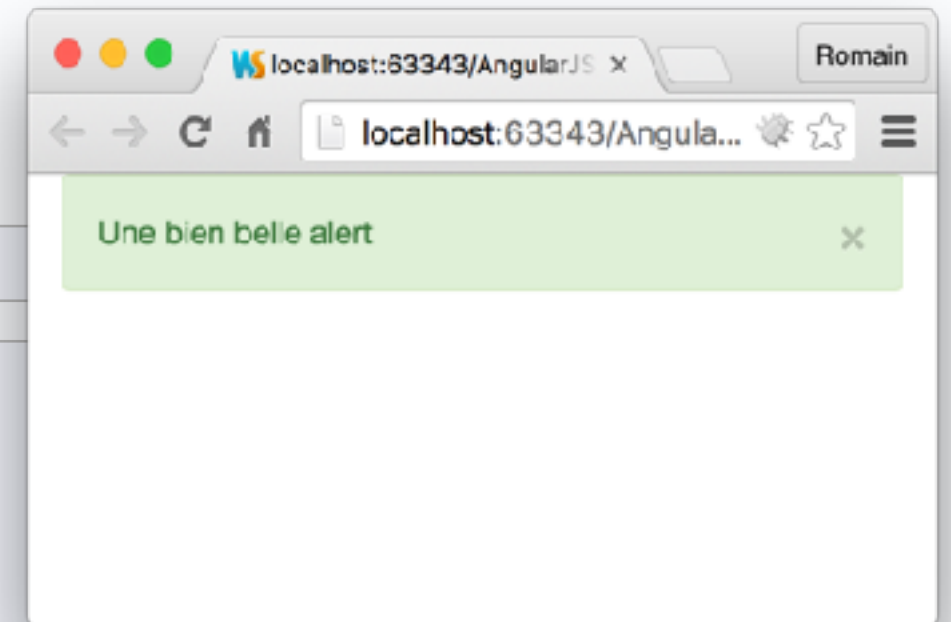
- ▶ Permet la création de balises ou d'attributs personnalisés pour simplifier le développement

Exemple : une alert bootstrap

```
<btp-alert>Une bien belle alert</btp-alert>
```

```
alertDirectives.directive('btpAlert', function() {  
  return {  
    transclude: true,  
    templateUrl: 'js/btp-alert.html'  
  };  
});
```

```
<div class="alert alert-success">  
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  
    <span aria-hidden="true">&times;</span>  
  </button>  
  <div ng-transclude></div>  
</div>
```



- ▶ ngTransclude permet de spécifier le point d'insertion du contenu de <btp-alert>



- ▶ **Contrôleur**

Facilite le découplage dans une application Angular.

- ▶ **ng-controller**

La directive ng-controller permet de lier le contrôleur à l'application.

- ▶ **A son propre scope descendant de celui de ng-app**

Les variables définies dans le contrôleur n'existent que dans celui-ci.

- ▶ **Déclaration dans un module**

Les modules regrouperont à terme d'autres types de composants

- ▶ **Bonnes pratiques**

Le contrôleur ne manipule pas le DOM directement, il se concentre sur la logique applicative.

Pas de filtrage, formatage ou de validation.



- Ici la variable `contacts` et la fonction `ajouter` deviennent disponibles dans la vue

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

addressBookModule.controller('AddressBookCtrl', ['$scope', function($scope) {
  $scope.contacts = [{
    prenom: 'Thierry',
    nom: 'Henry'
  }, {
    prenom: 'Zinédine',
    nom: 'Zidane'
  }];

  $scope.ajouter = function() {
    $scope.contacts.push({prenom: $scope.prenomSaisi, nom: $scope.nomSaisi});
  };
}]);
```

- Injection de dépendance
Dans cet exemple l'objet `$scope` est reçu automatiquement



► Déclaration d'un Service pour accéder aux données

```
var phonecatServices = angular.module('phonecatServices', ['ngResource']);

phonecatServices.factory('Phone', ['$resource',
  function($resource){
    return $resource('data/:phoneId.json', {}, {
      query: {method: 'GET', params: {phoneId: 'phones'}, isArray: true}
    });
  }
]);
```

► Injection dans le contrôleur et utilisation du Model

```
phonecatControllers.controller('PhoneListCtrl', ['$scope', 'Phone',
  function($scope, Phone) {
    $scope.phones = Phone.query();
    $scope.orderProp = 'age';
  }
]);
```



► Route

Angular permet la mise en place de routes, des URLs configurées permettront donc d'accéder à certains contrôleurs

```
var phonecatApp = angular.module('phonecatApp', [  
  // ...  
]);  
  
phonecatApp.config(['$routeProvider', '$locationProvider',  
  function($routeProvider, $locationProvider) {  
    $locationProvider.html5Mode(true);  
  
    $routeProvider.  
      when('/phones', {  
        templateUrl: 'partials/phone-list.html',  
        controller: 'PhoneListCtrl'  
      }).  
      when('/phones/:phoneId', {  
        templateUrl: 'partials/phone-detail.html',  
        controller: 'PhoneDetailCtrl'  
      }).  
      otherwise({  
        redirectTo: '/phones'  
      });  
  });  
});
```

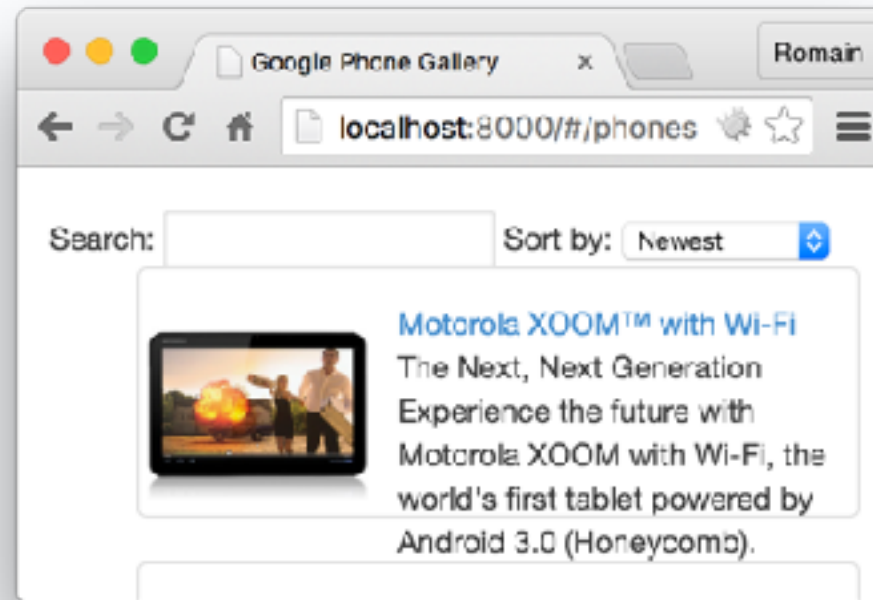
```
<!doctype html>  
<html lang="en" ng-app="phonecatApp">  
<head>  
  <meta charset="utf-8">  
  <title>Google Phone Gallery</title>  
  <base href="/">  
  ...  
</head>  
<body>  
  <div class="view-container">  
    <div ng-view></div>  
  </div>  
</body>  
</html>
```

AngularJS - Router

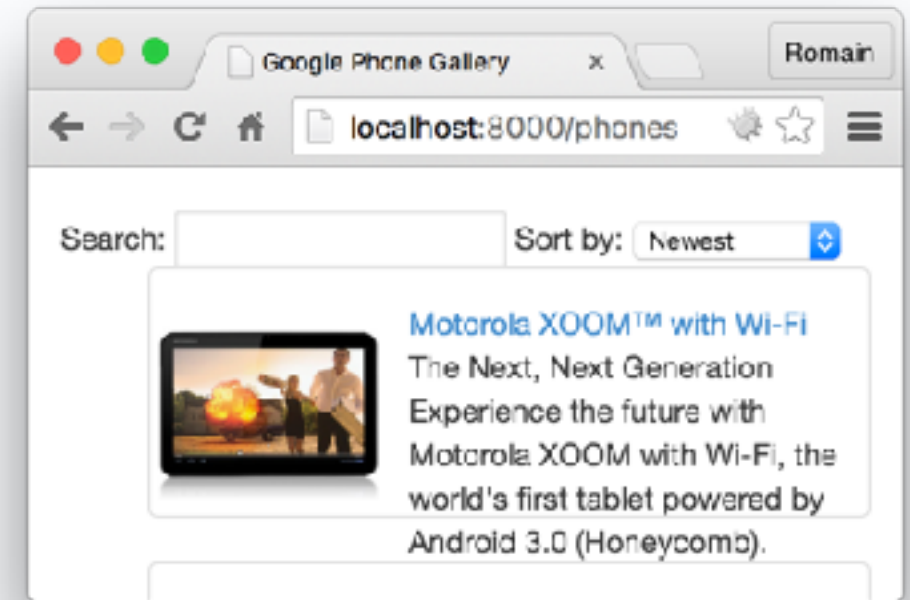


► 2 modes

Hash



HTML5



```
var express = require('express');
var app = express();

app.use('/bower_components', express.static(__dirname + '/app/bower_components'));
app.use('/css', express.static(__dirname + '/app/css'));
app.use('/img', express.static(__dirname + '/app/img'));
app.use('/js', express.static(__dirname + '/app/js'));
app.use('/partials', express.static(__dirname + '/app/partials'));
app.use('/data', express.static(__dirname + '/app/data'));

app.all('/*', function(req, res, next) {
  // Just send the index.html for other files to support HTML5Mode
  res.sendFile('app/index.html', { root: __dirname });
});

app.listen(8000); //the port you want to use
```



- Angular Batarang (Extension Chrome Dev Tools)
<https://github.com/angular/batarang>
- ng-inspector (Extension Chrome Dev Tools)
<http://ng-inspector.org/>
- Angular Hint (Script à installer)
<https://github.com/angular/angular-hint/>
- En utilisant les Dev Tools du Navigateur



► Angular Batarang

The screenshot shows a web browser window with the URL `localhost:63343/AngularJS/Controller/app.html#`. The page displays a list of contacts under the heading "Liste des contacts". The list contains two items: "Romain Bohdanowicz" and "Jean Dupont", each with a "supprimer" link. The AngularJS Batarang debugging interface is open, showing the "AngularJS" tab. The "Elements" panel on the left shows a tree view with "Enable" (checked), "Scopes", "Hints", and "Performance (Alpha)". The "Sources" panel on the right shows the source code of the application, with the following code highlighted:

```
<ng-app="monApp">
  <ng-controller="MonController">
    <ng-repeat="contact in contacts">
      <ng-repeat="contact in contacts">
```

The "Models" panel on the right shows the current model state:

```
{
  $id: 6,
  contact: {
    $index: 1,
    $first: false,
    $last: true,
    $middle: false,
    $even: false,
    $odd: true
  }
}
```

The "Console" panel at the bottom shows the "Search" tab selected, with the "Emulation" and "Rendering" tabs also visible. The "Console" panel shows the "top frame" selected, with a "Preserve log" checkbox and a blue arrow icon.

AngularJS - Déboguage



The screenshot shows a web browser at `localhost:63343/AngularJS/Controller/app.html#` displaying a contact list titled "Liste des contacts". The list contains two entries: "Romain Bohdanowicz" and "Jean Dupont", each with a "(supprimer)" link. The browser's developer tools are open, showing the "Elements" panel with the selected `` element. The "Styles" panel shows the default user agent styles for a list item. The "Console" panel displays the AngularJS scope object, which is a `ChildScope` instance. The scope object contains various properties, including `$$childTail`, `$$childHead`, `$$nextSibling`, `$$listeners`, `$$listenerCount`, `$$prevSibling`, `$$transcluded`, `$$watchers`, `$$watchersCount`, `seven`, `sfirst`, `sid`, `sindex`, `slast`, `smiddle`, `sodd`, `sparent`, `contact`, and `__proto__`.

Liste des contacts

- Romain Bohdanowicz [\(supprimer\)](#)
- Jean Dupont [\(supprimer\)](#)

Elements Network Sources Timeline Profiles Resources »

```
<!-- ngRepeat: contact in contacts -->
<li ng-repeat="contact in contacts" class="ng-binding ng-scope">
  "
  Romain Bohdanowicz
  "
  <a href="#" ng-click="supprimer($index)" />(supprimer)</a>
html.ng-scope body div.ng-scope ul li.ng-binding.ng-scope
```

Styles Event Listeners DOM Breakpoints Properties \$scope

Filter

```
element.style {
}
li {
  display: list-item;
  text-align: -webkit-match-parent;
}
```

Console Search Emulation Rendering

<top frame> ▾ ☐ Preserve log

```
> $scope
< ChildScope {$$childTail: null, $$childHead: null, $$nextSibling: ChildScope, $$watchers:
Array(3), $$listeners: Object...}
  $$childScope: null
  $$childHead: null
  $$childTail: null
  $$listenerCount: Object
  $$listeners: Object
  $$nextSibling: ChildScope
  $$prevSibling: null
  $$transcluded: true
  $$watchers: Array(3)
  $$watchersCount: 3
  seven: true
  sfirst: true
  sid: 5
  sindex: 0
  slast: false
  smiddle: false
  sodd: false
  sparent: ChildScope
  contact: Object
  __proto__: ChildScope
```

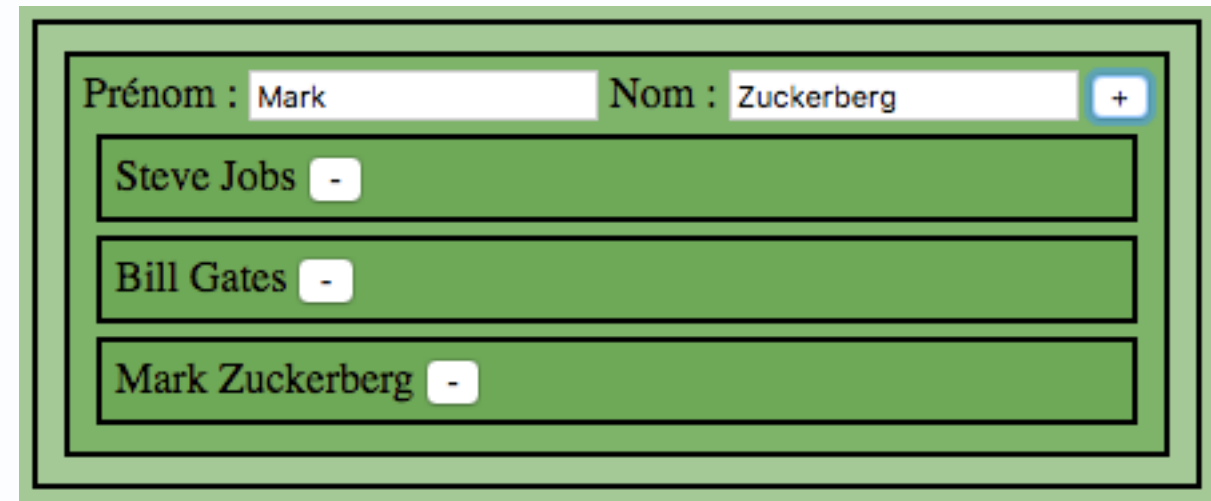


▸ Chrome Dev Tools

Il n'est pas nécessaire d'utiliser une extension

▸ En CSS

```
<style>
  .ng-scope {
    background-color: rgba(97, 161, 72, 0.57);
    margin: 5px;
    padding: 5px;
    border: 2px solid;
  }
</style>
```



▸ Dans la console

```
angular.element($0).scope()
```



- ▶ Tests Manuels

Manuellement il est possible de tester son application soit entre appelant manuellement certaines méthodes, soit en exécutant l'ensemble de l'application et en jouant un scénario de test.

- ▶ Tests Automatisés

Quand l'application grandit, il devient impensable de refaire tous ces tests manuellement, on cherche donc à les automatiser.

- ▶ AngularJS permet d'automatiser 2 types de tests

- ▶ Tests unitaires

- Testent de manière isolée chaque composant de l'application (contrôleurs, directives, filtres...)

- ▶ Les tests End to End (e2e)

- Testent de manière isolée chaque composant de l'application (contrôleurs, directives, filtres...)



- Les tests unitaires dans AngularJS dépendent de 3 outils
 - Un lanceur de test : Karma, qui va lancer les tests dans différents navigateurs et générer différents formats de rapports
 - Un framework de test : Jasmine, Mocha, qUnit ou autre, l'API dans lequel s'écrit le test
 - Un composant pour simplifier leur écriture : ngMock, qui contient entre autre des méthodes pour exécuter indépendamment des composants et des objets de tests permettant de simuler le comportement de ce qui est extérieur à l'application (requête
- Installation

```
npm install -g karma-cli
npm install karma jasmine-core karma-jasmine karma-chrome-launcher karma-jasmine-html-reporter --save-dev
```



- ▶ Jasmine

<http://jasmine.github.io/>

```
describe('Test de la classe Calculette', function() {  
  var calc;  
  
  beforeEach(function() {  
    calc = new Calculette();  
  });  
  
  describe('Test de la methode addition', function() {  
  
    it('1 + 1 devrait faire 2', function() {  
      expect(calc.addition(1, 1)).toEqual(2);  
    });  
  
  });  
});
```

- ▶ describe() : une suite de tests (un groupe de tests)
- ▶ it() : un test
- ▶ expect() : la comparaison du résultat et du résultat attendu
- ▶ beforeEach() : s'exécute avant chaque test



► Karma

<https://karma-runner.github.io/>

```
// karma.conf.js
module.exports = function(config) {
  config.set({

    basePath: './app',

    files: [
      'bower_components/angular/angular.js',
      'bower_components/angular-mocks/angular-mocks.js',
      'todolist/**/*.js'
    ],

    autoWatch: true,

    frameworks: ['jasmine'],

    browsers: ['Chrome'],

    reporters: ['progress', 'kjhtml'],

    plugins: [
      'karma-chrome-launcher',
      'karma-jasmine',
      'karma-jasmine-html-reporter',
    ],
  });
};
```



► Tester un contrôleur unitairement

```
// app/todolist/todolist.js
angular.module('app.todolist', [])

.controller('TodolistController', function($scope) {
  $scope.todolist = [];

  $scope.addTodo = function(todo) {
    $scope.todolist.push({
      value: todo.value
    });
  };
});
```

- L'appel de la méthode addTodo devrait ajouter un élément au tableau
- L'élément ajouté au tableau devrait contenir la même valeur que celle passée à addTodo



► Tester un contrôleur unitairement

```
// app/todolist/todolist_test.js
describe('app.todolist module', function() {

  beforeEach(module('app.todolist'));

  describe('TodolistController', function(){

    it('should add a todo when clicking plus button', inject(function($controller) {

      var $scope = {};
      var todolistCtrl = $controller('TodolistController', {$scope: $scope});
      expect(todolistCtrl).toBeDefined();

      $scope.addToDo({value: 'Learn Angular'});
      expect($scope.todolist.length).toBe(1);
      expect($scope.todolist[0].value).toBe('Learn Angular');

    }));

  });

});
```


AngularJS - Style Guide



▸ Extraits de :

<https://github.com/mgechev/angularjs-style-guide/blob/master/README-fr-fr.md>

▸ Arborescence :

