



<https://angularjs.org/>



## Le Framework AngularJS <https://angularjs.org/>

- Framework javascript depuis 2009
- Aujourd'hui (wikipedia) le framework est utilisé dans >8,400/1,000,000 sites web
- Open-source (MIT licence)
- Répond à deux architectures:
  - MVC : model – view - controller
  - MVVM : model – view – viewmodel

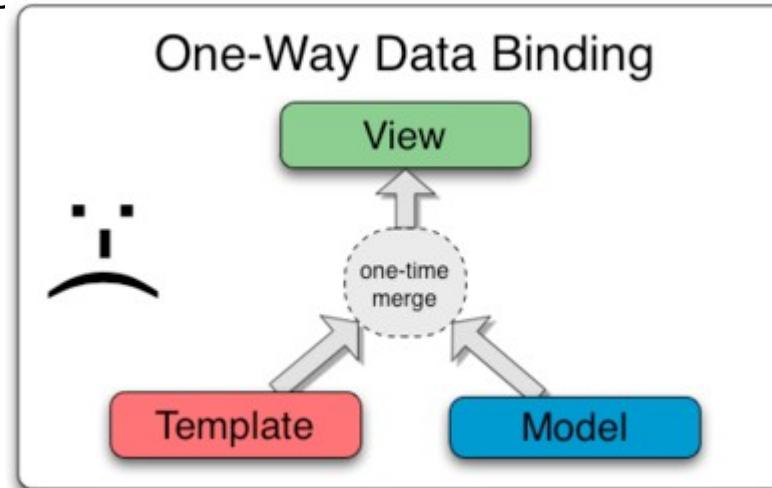
## Philosophie de AngularJS

- Séparation de la vue de la logique métier :
  - La vue est un **template** HTML (comparable à du JSP) qui est compilé par le browser
  - Le métier (ajax..) est du code javascript qui **pilote** le comportement de la vue HTML
    - la vue est entièrement construite en HTML (et non construite en javascript)
- Des attributs (**ng-xxx**) permettent l'affichage de la page ; elles permettent de partager :
  - Des variables
  - Des conditions (ex : affichage ou non d'une div, répétition selon une liste, ...)
  - Des événement (ex : événements liés aux formulaires clique sur un bouton, etc)
- Utilise le « **data binding** » pour synchroniser le modèle et la vue (~trigger)

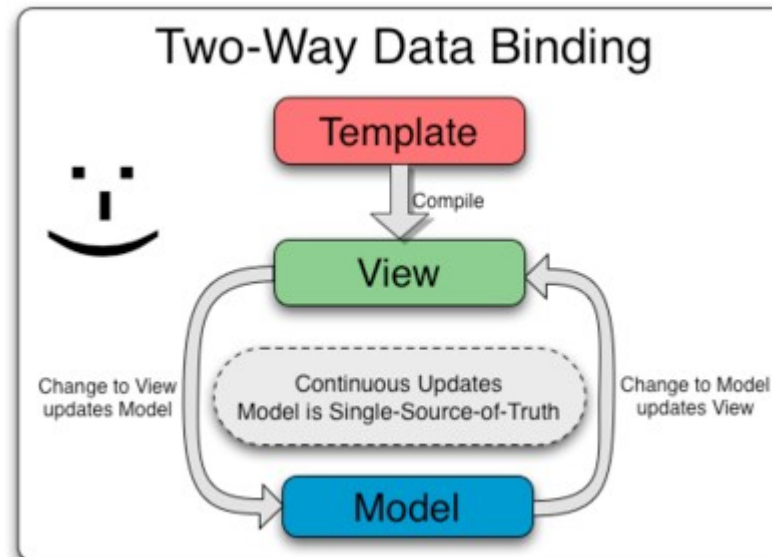
## Le data binding : AngularJS propose une architecture 2-ways Data-binding

- Cas classique : la vue est générée depuis le template et le modèle

Ex : Le model (code JavaScript) agit sur la vue à partir des élément de celle-ci



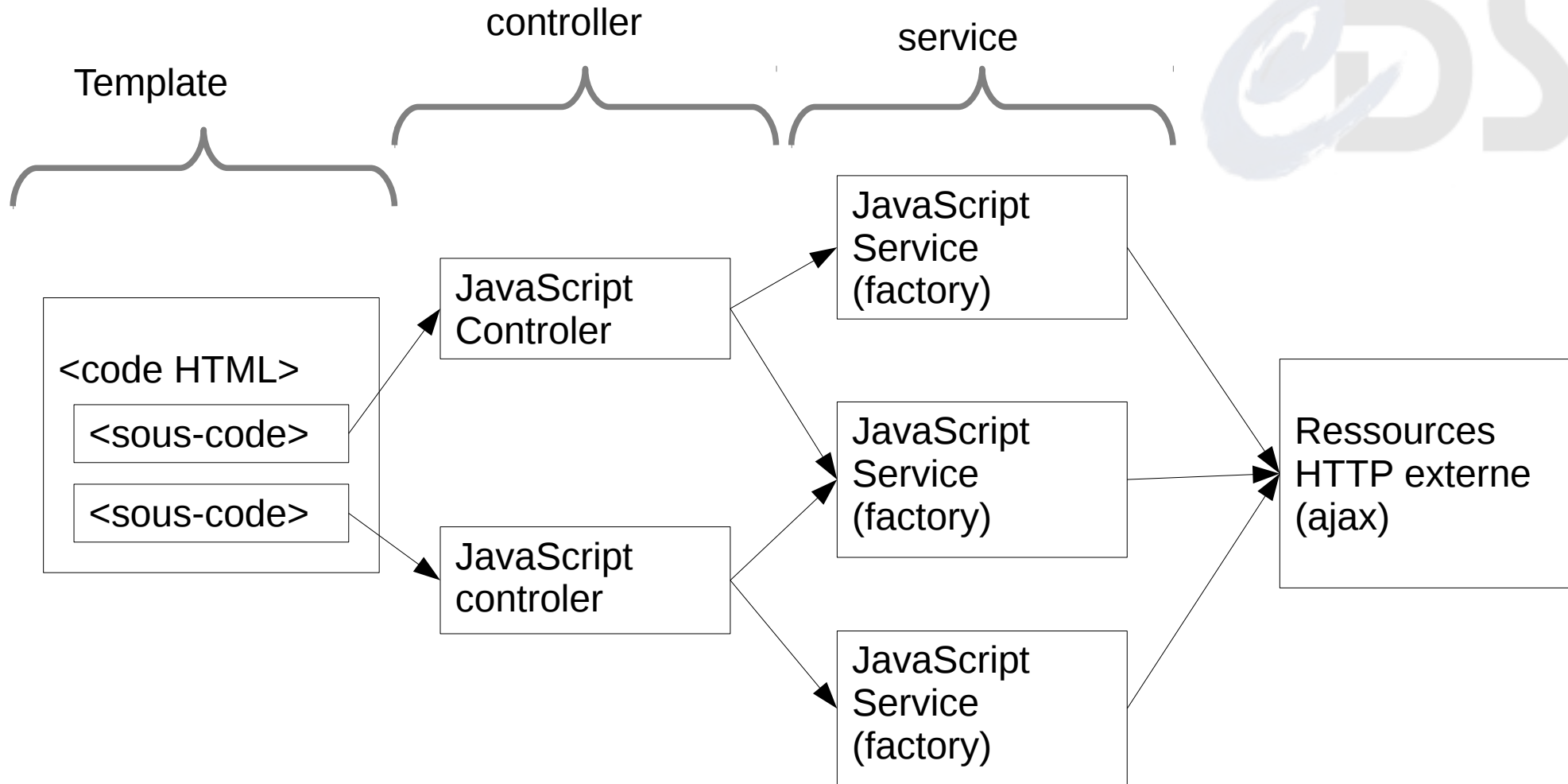
- AngularJS : le template est compilé en une vue dynamique. La vue modifie le modèle et réciproquement





## Vocabulaire et notions AngularJS

- Le **model** : Il s'agit d'un jeu de données avec lequel interagit Page HTML et code Javascript. il est défini par les attributs **ng-model** des éléments du template HTML
- Le **controller** : contrôle les événements du template HTML :
  - il s'applique aux événements du formulaire (boutons..) et plus généralement à tout élément du template dont un model ou événement est affecté
  - Aux variables et expressions angularJS du template HTML
- Le **module** : il s'applique à un élément du template HTML : il comprend le controller , les filtres (plusieurs modules (ou application) peuvent être défini dans un template)
- Le **scope** : il s'agit du contexte incluant contrôleurs, variables, module, services qui sont accessible depuis la vue (template HTML)
- Le **service** : ou factory définit une sorte de module regroupant une même logique (ex : une factory AladinLite, Acces aux données (Ajax), Verifications de formulaire..)





## Premiers exemples

```

<html>
<head>
  <script src='angular.min.js'></script>
</head>
<body ng-app="myApp" ng-controller="myctl">
  Author {{name}}<br/>
  <button ng-click="next()">Counter</button><br/>
  you pushed {{count}} times
</body>

```

```

<script type='text/javascript'>
var application = angular.module("myApp", []);
application.controller('myctl', function($scope) {
  $scope.name="Gilles Landais";
  $scope.count = 0;
  $scope.next = function () { $scope.count ++; };
});
</script>
</html>

```

Variables  
du scope

→ \$scope.name="Gilles Landais";  
 → \$scope.count = 0;  
 \$scope.next = function () { \$scope.count ++; };

Création du  
module « myApp »

Classe controller  
« myctl »



```
<body ng-app="myApp" >
```

```
  Choose page
```

```
  <select ng-model='page'>  
    <option value='1'>1</option>  
    <option value='2'>2</option>  
  </select>
```

Model « page » est commun  
au code et au template

```
  <div ng-show='page==1'>  
    <span style='color: red'>Page 1</span>  
  </div>
```

Événement pour la  
gestion de l'affichage

```
  <div ng-show='page==2'>  
    <span style='color: blue'>Page 2</span>  
  </div>  
</body>
```

```
<script type='text/javascript'>  
  angular.module("myApp", []);  
</script>
```



```
<body ng-app="myApp" ng-controller="myctl">
```

```
  Choose page
```

```
  <select ng-model='page'>
```

```
    <option value='1'>1</option>
```

```
    <option value='2'>2</option>
```

```
</select>
```

```
<div ng-show='page==1'>
```

```
<span style='color: red'>Page 1</span>
```

```
</div>
```

```
<div ng-show='page==2'>
```

```
<span style='color: blue'>Page 2</span>
```

```
</div>
```

```
</body>
```

```
<script type='text/javascript'>
```

```
angular.module("myApp", [])
```

```
  .controller("myctl", function($scope) {
```

```
    $scope.page=2;
```

```
    $scope.$watch("page", function() {
```

```
      alert("Change page -> "+$scope.page);
```

```
    });
```

```
  });
```

```
</script>
```

Ajout d'un  
« espion » sur le  
model « page »  
pour exécuter une  
fonction







```
<body ng-app="myApp" ng-controller="myctl">
  <select ng-model="action">
    <option value='1'>START</option>
    <option value='2'>STOP</option>
  </select>
  <button ng-click="execute()">Go</button>
  <div ng-show="state.length >0">
    <ul><li ng-repeat="item in state">{{item}}</li></ul>
  </div>
</body>
```

```
<script type='text/javascript'>
angular.module("myApp", [])
  .controller('myctl', function($scope, automate) {
    $scope.execute = function () {
      if ($scope.action == 1) $scope.state = automate.start();
      else if ($scope.action == 2) $scope.state = automate.stop();
    };
  })
  .factory("automate", function() {
    var factory = {};
    factory.start = function() { return ["START", (new Date()).toString()]; }
    factory.stop = function() { return ["STOP", (new Date()).toString()]; }
    return factory;
  });
</script>
```

Ajout de la factory  
comme paramètre du  
controlleur

Définition  
de la  
factory

# Utilisation d'un filtre et d'une requete Ajax



```
<html>
<head>
  <script src='angular.min.js'></script>
  <script src='x2js-v1.0.11/xml2json.js'></script>
</head>
<body ng-app="myApp" ng-controller="myctl">
  Hipparcos, position: <input type='text' ng-model='position' value='0+0' />
  <button ng-click="queryvizier()">GO</button>

  <table>
    <th ng-repeat="field in vizier.fields" title="{{field.DESCRPTION}}">{{field._name}}</th>
    <tr ng-repeat="row in vizier.table">
      <td ng-repeat="td in row.TD">{{td| round }}</td>
    </tr>
  </table>
</body>

<script type='text/javascript'>
angular.module("myApp", [])
.controller("myctl", function($scope, $http, vizier) {
  $scope.position="0+0";
  $scope.vizier=vizier;
  $scope.queryvizier = function() {
    vizier.query($http,$scope.position);
  };
})
.factory("vizier", function($http) {
  var vizier={};
  vizier.query = function ($http,position) {
    $http.get("http://vizier.u-strasbg.fr/viz-bin/votable?-source=1239/hip_main&-c.rd=2&-c="+position)
    .success(function(response) {
      var x2js = new X2JS();
      var json = x2js.xml_str2json( response );
      vizier.table = json.VOTABLE.RESOURCE.TABLE.DATA.TABLEDATA.TR;
      vizier.fields = json.VOTABLE.RESOURCE.TABLE.FIELD;
    });
  };
  return vizier;
})
.filter('round', function() {
  return function(input, total) {
    if (input instanceof Object) return;
    f = Math.round(input*100)/100;
    return f ? f: input;
  };
});
</script>
</html>
```



## Mais aussi ...

- Les tables (par exemple issues d'une requête Ajax) sont construites dans le template HTML avec l'attribut **ng-repeat**
- Les **filtres** sont d'autres object AngularJS qui permettent d'appliquer des propriétés aux tables : sort, pagination, ...
- Des objets ajoute de l'etendue au code AngularJS : \$http (Ajax) , \$watch , \$interval (timer)...
- Il existe des extensions : bootstrap, Animations, REST API, etc.
- Il est possible d'importer des modules, factory externes...

## Conclusions

- AngularJS est un framework qui permet de bien séparer vue du model en offrant une vision simple de la vue
- Demande un apprentissage et n'ai pas lisible pour le néophite
- Les erreurs de syntaxes ne sont pas toujours très explicite dans les traces d'erreurs