

I. Programmation AngularJS

Table des matières

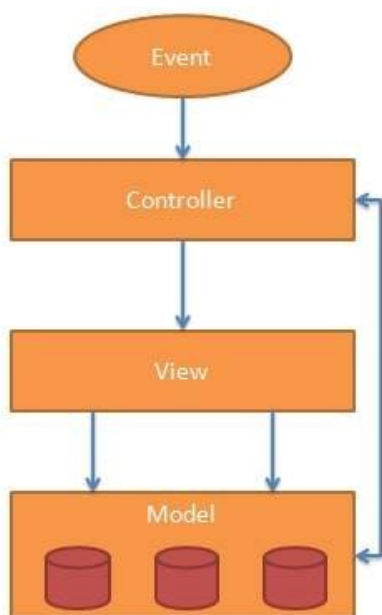
I. Programmation AngularJS.....	1
AngularJS.....	2
1. Architecture et modèle d'application.....	2
Architecture MVC.....	2
Modèle de code.....	2
Application minimale.....	3
2. Directives.....	3
Directives de Angular.....	4
Créer de directives.....	4
3. Expressions.....	5
4. Contrôleur.....	6
5. Les filtres.....	7
Introduction.....	7
Définir son propres filtre.....	8
Tables et itération.....	8
6. Gestion de la DOM.....	8
7. Modules.....	9
8. Formulaire.....	9
Événements avec \$event.....	10
9. Include.....	11
10. Scope.....	11
11. Services.....	12
Introduction.....	12
Service http et scope.....	12
Créer son propre service.....	13
Routage.....	14
Timer.....	15
Canvas.....	16
Video.....	18
Géolocalisation.....	18

AngularJS

1. Architecture et modèle d'application

Architecture MVC

Angular JS est un design pattern de type MVC permettant d'organiser le code Javascript côté navigateur.



AngularJs introduit des tags attributs permettant de mettre en place une facilité de codage. Il est compatible avec les standards : javascript, jquery, html, css, ajax, ...

Fonctionnalités :

- Directives
- Filtres
- Services
- Vues
- Formes
- Expression
- Tags spécifiques
- Scope
- Injection de dépendance

Modèle de code

Inclure la librairie

```
| <head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0-beta.17/angular.min.js"></script>
</head>
```

Vues avec des tags et expressions

```
<div ng-app = "mainApp" ng-controller = "CalcController">
  <p>Enter a number: <input type = "number" ng-model = "number" /></p>
  <button ng-click = "square()">X<sup>2</sup></button>
  <p>Result: {{result}}</p>
</div>
```

Controleur et modèle

```
mainApp.controller('CalcController', function($scope, CalcService, defaultInput) {
  $scope.number = defaultInput;
  $scope.result = CalcService.square($scope.number);
  $scope.square = function() {
    $scope.result = CalcService.square($scope.number);
  }
});
```

Application minimale

```
<!doctype html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0-beta.17/angular.min.js"></script>
  </head>
  <body ng-app="monapp">
    <div ng-controller="monController" >
      <h1>Je dis {{bonjour}} à {{aqui}}</h1>
    </div>
    <script>
      angular.module("monapp", [])
        .controller("monController", function($scope) {
          $scope.bonjour = "bonjour";
          $scope.aqui = " quelqu'un";
        });
    </script>
  </body>
</html>
```

2.Directives

Pour étendre les fonctionnalité de HTML, Angular possède ses propres directives

Directives de Angular

- ng-app : début de l'application
- ng-init : initialisation des données de l'application
- ng-controller : définitions du contrôleur
- ng-model : binding entre AngularJS et les <input de HTML
- ng-repeat : faire une itération sur une collection

```
<body ng-app="monapp">
  <div ng-controller="monController1 as c1" >
    <span ng-repeat="d in data | filter:reverse">{{d+'--'}}</span>
  </div>
  <script>
    angular.module("monapp", [])
      .controller("monController1", function($scope) {
        $scope.data = [10,20,30,40,50,'AA','BB','CC'];
      });
  </script>
</body>
```

```
<p>Nom: <input type = "text" ng-model = "nom"></p>
```

Créer de directives

On peut créer sa propre directive de type

- Element directives
- Attribute

```

<body>
  <h2>Les directives</h2>
  <div ng-app="mainApp">
    <personne prenom="Karim"></personne><br/>
  </div>
  <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"></script>
  <script>
    var mainApp = angular.module("mainApp", []);

    mainApp.directive('personne', function() {
      var directive = {
        restrict : 'E',
        scope : {
          prenom : '=prenom'
        },
        compile : function(element, attrs) {
          element.css("border", "1px solid #cccccc");
          element.css("font-size", "22px");
          var linkFunction = function($scope, element, attrs) {
            element.html(attrs.prenom);
            element.css("background-color", "#ffff00");
          }
          return linkFunction;
        }
      };
      return directive;
    });

  </script>
</body>

```

3. Expressions

Permettent de faire le lien entre HTML les données pour afficher leur contenu ou bien évaluer une expression comme un opération arithmétique ...

Le symbole est {{ }}

```

<div ng-init="quantite=10;prix=30">
  <h2>Prix total</h2>
  <p> Cout du livre : {{prix * quantite}} €</p>
  <table border="1" ng-init="totos=[

```

```
{nom:'TOTO1',prenom:'toto1', age:1},
{nom:'TOTO2',prenom:'toto2', age:2},
{nom:'TOTO3',prenom:'toto3', age:3},
{nom:'TOTO4',prenom:'toto4', age:4}]]">
<tr ng-repeat="tt in tolos" >
  <td ng-repeat="t in tt">
    {{ t }}
  </td>
</tr>
</table>
<table border="1">
<tr ng-repeat="tt in tolos" >
  <td ng-repeat="t in tt">
    {{ Math.round(Math.random()*100)}}
  </td>
</tr>
</table>
<table border="1">
<tr ng-repeat="tt in tolos" >
  <td ng-repeat="t in tt">
    {{ rand(100)}}
  </td>
</tr>
</table>
</div>
```

4. Contrôleur

Les contrôleur permettent de gérer les échanges entre le modèle et les vues.

Les données peuvent venir de la vues par une saisie par exemple ou bien venir du modèle d'une requête Ajax par exemple.

La synchronisation implicite des données entre vue et modèle se fait au moyen de \$scope

```
<body ng-app="monapp">
  <div ng-controller="monController1 as c1" >
    <span ng-repeat="d in data | filter:reverse">{{ d+'--' }}</span>
  </div>
<script>
  angular.module("monapp", [])
    .controller("monController1", function($scope) {
      $scope.data = [10,20,30,40,50,'AA','BB','CC'];
```

```

    });
</script>
</body>

```

Plus complet avec une requête http

```

<body ng-app="monapp">
  <button ng-click='$scope.data=[]>Raffraichir</button>
  <div ng-controller="monController" >
    <span ng-repeat='d in data.info'>
      <input ng-model='d'/></br>
    </span>
    La liste : <span ng-bind="data.info"/><br/>
    {{data.info}}
  </div>
<script>
  var m=angular.module("monapp", [])
  m.controller("monController", function($scope,$http) {
    $scope.data = {
      info : [],
      affiche : function() {
        return $scope.data.info;
      }
    };
    var rep = $http.get("/angularjs/controleur/server.php");
    rep.success(function(data, status, headers, config) {
      console.log(rep);
      $scope.data.info = data;
    });
    rep.error(function(data, status, headers, config) {
      console.log("Probleme");
    });
  });
</script>
</body>

```

5. Les filtres

Introduction

Les filtres utilisent le symbole de pipe (|) pour transformer les données. Il existe de nombreux filtres et on peut en définir

```
{{ expression | filter:argument1:argument2:... }}
```

- lowercase
- uppercase
- currency

Contenu : {{data | lowercase}}

Definir son propres filtre

```
<body ng-app="monapp">
  <div ng-controller="monController1 as c1" >
    <span ng-repeat="d in data">{{d|formatage}}</span>
  </div>
  <script>
    angular.module("monapp", [])
      .filter('formatage', function() {
        return function (ladata){
          return (res = ladata + '***');
        }
      })
      .controller("monController1", function($scope) {
        $scope.data = [10,20,30,40,50,'AA','BB','CC'];
        console.log($scope.data)
      });
  </script>
</body>
```

Tables et itération

On peut utiliser ng-repeat pour remplir une table HTML

```
<table>
  <tr>
    <th>Name</th>
    <th>Marks</th>
  </tr>

  <tr ng-repeat = "p in personne">
    <td>{{ p.getAll() }}</td>
  </tr>
</table>
```

6. Gestion de la DOM

Pour agir sur les contrôles HTML on manipule la DOM associée

- ng-disabled
- ng-show
- ng-hide
- ng-click

```
<p>Total click: {{ compteur }}</p>
<button ng-click = "compteur=compteur + 1">Click</button>
```

7. Modules

AngularJS est modulaire et permet de séparer physique les différentes parties de l'application

- Services
- Application
- Contrôleurs

main.js

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

moncontrôleur.js

```
mainApp.controller("monContrôleur", function($scope) {
})
```

```
<div ng-app = "mainApp" ng-controller = "monContrôleur">
  <script src = "mainApp.js"></script>
  <script src = "moncontrôleur.js"></script>
</div>
```

8. Formulaire

Les formulaires constituent la partie vue du MVC. Angular utilise des directives pour synchroniser les données du modèle avec le contenu du formulaire avec ng-modèle ou {{ }}

Événements :

```
ng-click
ng-dbl-click
ng-mousedown
ng-mouseup
ng-mouseenter
ng-mouseleave
ng-mousemove
ng-mouseover
ng-keydown
ng-keyup
```

ng-keypress

ng-change

```
<body ng-app="monapp">
  <div ng-controller="monController" >
    <select ng-model="selection"
      ng-options="f as f.label for f in fruits">
    </select>
    <p>Fruit courant: {{selection.label}} : {{selection.value}}</p>
  </div>
  <script>
    angular.module("monapp", [])
      .controller("monController", function($scope) {
        $scope.fruits = [
          { label : 'oranges', value : 1 },
          { label : 'raisins', value : 2 },
          { label : 'bananes', value : 4 },
          { label : 'kiwi', value : 5 },
          { label : 'poires', value : 6 }
        ];
        $scope.selection = $scope.fruits[1];
      });
  </script>
</body>
```

Événements avec \$event

```
<body ng-app="myapp">
  <div ng-controller="MyController" >
    <div ng-click="myData.doClick($event)">Click here</div>
    <div ng-bind="myData.coord">X+Y</div>
  </div>

  <script>
    angular.module("myapp", [])
      .controller("MyController", function($scope) {
        $scope.myData = {};
        $scope.myData.doClick = function(event) {
          alert("clicked: " + event.clientX + ", " + event.clientY);
          $scope.myData.coord="clicked: " + event.clientX + ", " +
event.clientY;
        }
      });
  </script>
</body>
```

```
        } );  
  
</script>  
</body>
```

9. Include

Permet d'inclure des fichiers extérieurs

```
<div ng-app = "" ng-controller = "monControlleur">  
  <div ng-include = "main.htm"></div>  
  <div ng-include = "personne.htm"></div>  
</div>
```

10. Scope

Le Scope contient les objets (données et méthodes) devant être accessible à la vue dans le formulaires

```
<body ng-app="monapp">  
<div ng-init="" ng-controller="monController">  
  <h2>Formulaire</h2>  
  <form name="fpersonne" action="#" ng-submit="submitForm()">  
    Nom :<input ng-model='personne.nom'/><br/>  
    Prenom :<input ng-model='personne.prenom'/><br/>  
    Age :<input type='Number' ng-model='personne.age'/><br/>  
    <input type="submit" ngClick="Submit" >  
  </form>  
  Personne : {{affiche()}} <br/>  
  Resultat: {{res}} <br/>  
</div>  
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0-beta.17/angular.min.js"></script>  
<script>  
angular.module ('monapp', [])  
.controller("monController", function($scope, $http) {  
  $scope.personne={  
    nom:'vide',  
    prenom:'vide',  
    age:-1  
  };  
  $scope.res="";  
  $scope.affiche = function ()  
  {  
    return $scope.personne.nom + "--" + $scope.personne.prenom + "--" + $scope.personne.age;
```

```
}  
$scope.save = function() {  
    $scope.personne = $scope.fpersonne;  
};  
  
$scope.submitForm = function() {  
    console.log($scope.personne);  
    $scope.personne = $scope.fpersonne;  
    $http.post('save.php', JSON.stringify($scope.personne))  
        .success(function(){  
            $scope.res="OK";  
        });  
};  
});  
</script>  
</body>
```

11. Services

Introduction

Le contrôleur utilise des services pour accéder à des fonctionnalités supérieures. Ils sont passés en argument au contrôleur pour coder (injection)

Un service encapsule un ensemble de fonctionnalités. Il existe des services offerts par AngularJs et on peut créer son propre service.

- \$http
- \$scope
- \$location : search(), path
- \$interval

Service http et scope

utilisation de \$http

```
<script>  
angular.module('monapp', [])  
.controller("monController", function($scope, $http) {  
    $scope.personne={  
        nom:'vide',  
        prenom:'vide',  
        age:-1  
    };  
    $scope.res="";  
});
```

```
$scope.affiche = function ()
{
    return $scope.personne.nom + "--" + $scope.personne.prenom + "--" + $scope.personne.age;
}
$scope.save = function() {
    $scope.personne = $scope.fpersonne;
};

$scope.submitForm = function() {
    console.log($scope.personne);
    $scope.personne = $scope.fpersonne;
    $http.post('save.php', JSON.stringify($scope.personne))
    .success(function(){
        $scope.res="OK";
    });
};
});
</script>
```

Créer son propre service

```
<body>
<h2>AngularJS </h2>
<div ng-app="mainApp" ng-controller="CalcController">
    <p>Enter a number: <input type="number" ng-model="number" />
    <button ng-click="square()">X<sup>2</sup></button>
    <p>Result: {{result}}</p>
</div>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"></script>
<script>
var mainApp = angular.module("mainApp", []);
mainApp.factory('MathService', function() {
    var factory = {};
    factory.multiply = function(a, b) {
        return a * b
    }
    return factory;
});

mainApp.service('CalcService', function(MathService){
    this.square = function(a) {
        return MathService.multiply(a,a);
    }
});
});
```

```

    }
  });

  mainApp.controller('CalcController', function($scope, CalcService) {
    $scope.square = function() {
      $scope.result = CalcService.square($scope.number);
    }
  });
</script>
</body>

```

Routage

Permet de se positionner dans des emplacements du document html en décodant les URLs

index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Routage</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.5/angular.min.js"></script>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.2.5/angular-route.min.js"></script>
</head>

<body ng-app="monapp" ng-controller='RouteController'>
  <a href="#/data1/12">Route 1</a><br/>
  <a href="#/data2/23">Route 2</a><br/>
  YYYYY : {{param}}<br/>
  <ng-view></ng-view>
<script>
  var module = angular.module("monapp", ['ngRoute']);

  module.config(['$routeProvider',
    function($routeProvider) {
      $routeProvider.
        when('/data1/:param', {
          templateUrl: 'data1.html',
          controller: 'RouteController'
        }).
        when('/data2/:param', {
          templateUrl: 'data2.html',
          controller: 'RouteController'

```

```
    }).  
    otherwise({  
        redirectTo: '/'  
    });  
}]);  
  
module.controller("RouteController", function($scope, $routeParams) {  
    $scope.param = $routeParams.param;  
    console.log($scope.param)  
})  
  
</script>  
</body>  
</html>
```

data1.html

```
<html>  
<body style='background : yellow'>  
<h1>Data1</h1>  
XXXXXX : {{param}}  
</body>  
</html>
```

data2.html

```
<html>  
<body style='background : yellow'>  
<h1>Data2</h1>  
XXXXXX : {{param}}  
</body>  
</html>
```

Timer

Le timer est un services

```
<body>  
<h2>Timer</h2>  
<div ng-app="myapp" ng-controller="monController">  
</div>  
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"></script>  
<script>  
    var myapp = angular.module("myapp", []);  
    myapp.controller("monController", function($scope, $interval){
```

```
        $interval(callAtInterval, 500);
    });
    function callAtInterval() {
        console.log("Interval occurred");
    }
</script>
</body>
```

Canvas

```
<html>
<head>
<title>Angular JS Includes</title>
</head>
<body>
<h2>AngularJS Sample Application</h2>
<div ng-app="">
<canvas drawing></canvas>
</div>
<script>
app.directive("drawing", function(){
    return {
        restrict: "A",
        link: function(scope, element){
            var ctx = element[0].getContext('2d');

            // variable that decides if something should be drawn on mousemove
            var drawing = false;

            // the last coordinates before the current move
            var lastX;
            var lastY;

            element.bind('mousedown', function(event){
                if(event.offsetX!==undefined){
                    lastX = event.offsetX;
                    lastY = event.offsetY;
                } else { // Firefox compatibility
                    lastX = event.layerX - event.currentTarget.offsetLeft;
                    lastY = event.layerY - event.currentTarget.offsetTop;
                }
            })
        }
    }
})
```



```
// begins new line
ctx.beginPath();

drawing = true;
});
element.bind('mousemove', function(event){
  if(drawing){
    // get current mouse position
    if(event.offsetX!==undefined){
      currentX = event.offsetX;
      currentY = event.offsetY;
    } else {
      currentX = event.layerX - event.currentTarget.offsetLeft;
      currentY = event.layerY - event.currentTarget.offsetTop;
    }

    draw(lastX, lastY, currentX, currentY);

    // set current coordinates to last one
    lastX = currentX;
    lastY = currentY;
  }

});
element.bind('mouseup', function(event){
  // stop drawing
  drawing = false;
});

// canvas reset
function reset(){
  element[0].width = element[0].width;
}

function draw(lX, lY, cX, cY){
  // line from
  ctx.moveTo(lX,lY);
  // to
  ctx.lineTo(cX,cY);
  // color
  ctx.strokeStyle = "#4bf";
```

```
// draw it
ctx.stroke();
}
}
};
});
</script>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.15/angular.min.js"></script>
</body>
</html>
```

Video

```
<body>
<video width="320" height="240" id='myVideo' autoplay >
  <source src="movie.mp4" type="video/mp4" >
</video>
<input type='button' value='start' onclick='playVid()'>
<input type='button' value='stop' onclick='pauseVid()'>
<script language='javascript'>
function playVid() {
  var vid = document.getElementById("myVideo");
  vid.play();
}

function pauseVid() {
  var vid = document.getElementById("myVideo");
  vid.pause();
}
</script>
</body>
```

Géolocalisation

```
<!doctype html>
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0-beta.17/angular.min.js"></script>
    <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
  </head>
  <body ng-app="monapp">
    <div ng-init="" ng-controller="monController">
```

```

<div id='lidmap'>XXX</div>
Latitude      : {{position.coords.latitude}}<br/>
Longitude    : {{position.coords.longitude}}<br/>
<h2>{{erreur}}</h2>
</div>
<script>
angular.module ('monapp', [])
.controller("monController", function($scope) {
    console.log(navigator)
    $scope.erreur='-----';
    $scope.geo = {
        init : function ()
        {
            if (navigator.geolocation) {
                navigator.geolocation.getCurrentPosition(this.lesuccess,
this.lerreur);
            } else {
                $scope.erreur="Your browser does not support geolocation.";
            }
            $scope.position = {
                coords : {
                    latitude :99999,
                    longitude :99999
                }
            };
            $scope.erreur="Initialisation ok";
        },
        lesuccess : function (position) {
            $scope.position.coords.latitude = position.coords.latitude;
            $scope.position.coords.longitude = position.coords.longitude;
            console.log(position.coords.latitude)
            console.log(position.coords.longitude)
            $scope.erreur='Coords';
            //console.log($scope.geo.lamap);
            $scope.geo.lamap(position)
            $scope.$apply($scope.model);
        },
        lerreur : function () {
            $scope.erreur = "There was an error retrieving your location!";
        },
        lamap: function (position) {

```

```
        var coord = position.coords.latitude + "," + position.coords.longitude;
        var img_url = "http://maps.googleapis.com/maps/api/staticmap?center="
        +coord+"&zoom=14&size=400x300&sensor=false";
        $("#lidmap").html("<img src='"+img_url+"'>");
        $scope.$apply($scope.model);
    }
}
$scope.geo.init();
console.log("init")
});
</script>
</body>
</html>
```