

ANGULARJS

INTRODUCTION TO ANGULARJS

Created by [Binita Mehta](#) / [@binita_mehta](#)

binita.mehta@agilex.com

AGENDA

- Introduction
- Core Angular Concepts
 - View, Controller, \$scope, Expressions, Model, Two-way Data Binding Directives, Filters
 - Service
 - Forms
- SPA and Angular UI Router
- Conclusion
 - Resources
 - Questions

WHAT IS ANGULARJS?

- Structural framework for dynamic web apps
- Client side
- HTML as template language
- Backed by Google

WHY IS IT CALLED "ANGULAR"?

- HTML has angular brackets `<html>` `<body>`
- "ng" prefix sounds similar to angular (ng-model, ng-repeat, ng-show)

BENEFITS OF ANGULARJS

- MVC Done right
- Organized
- DOM Manipulations separated from business logic
- Unit Testing ready

INITIALIZING AN ANGULAR APP

```
<html>
<head>
</head>
<body ng-app="myApp">
<script src="bower_components/angular/angular.min.js"></script>
</body>
</html>
```

VIEW

DOM - what the user sees

```
<html>
<head>
</head>
<body ng-app="myApp">
  <div ng-controller="MyRecipeController1">
    <h1>{{caption}}</h1>
  </div>
  <script src="bower_components/angular/angular.min.js"></script>
  <script src="scripts/controllers/controller1.js"></script>
</body>
</html>
```

CONTROLLER

Business logic behind views

```
angular.module('myApp', [])  
.controller('MyRecipeController1', function($scope) {  
    $scope.caption="My Tasty Recipes"  
})
```


\$SCOPE

Glue between controller and view

```
angular.module('myApp', [])  
  .controller('MyRecipeController1', function($scope) {  
    $scope.caption="My Tasty Recipes"  
  })
```

```
<html>  
<head>  
</head>  
<body ng-app="myApp">  
  <div ng-controller="MyRecipeController1">  
    <h1>{{caption}}</h1>  
  </div>  
  <script src="bower_components/angular/angular.min.js"></script>  
  <script src="scripts/controllers/controller1.js"></script>  
</body>  
</html>
```

EXPRESSIONS

JavaScript-like code snippets that are usually placed in bindings such as `{{ expression }}`

- `{{1+2}}`
- `{{a+b}}`
- `{{name}}`

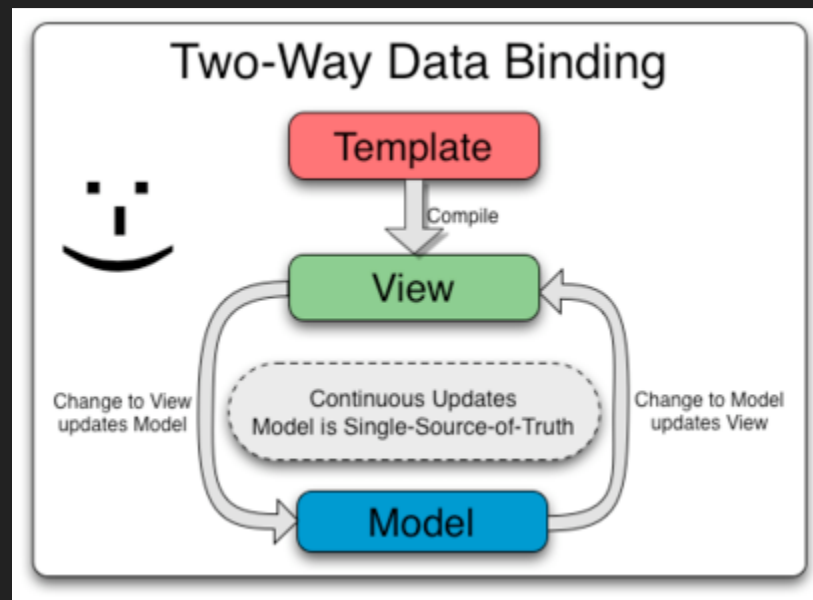
DEMO 1 - CONTROLLER, VIEW, \$SCOPE, EXPRESSIONS

MODEL

The data shown to the user in the view and with which the user interacts

```
angular.module('myApp', [])  
.controller('MyRecipeController2', function($scope) {  
    $scope.caption="My Tasty Recipes";  
    $scope.recipeName = "Chocolate Cake";  
})
```

TWO-WAY DATA BINDING



Source: docs.angularjs.org

DEMO 2: TWO-WAY DATA BINDING

DIRECTIVES

Custom Attributes and Elements that help extend HTML

ng-repeat, ng-show, ng-hide

DEMO 3: DIRECTIVES

FILTERS

- Formats the value of an expression
- Format {{expression | filter}}. Example {{2 | currency}} will show \$2.00
- Can be used in view templates, controllers or services
- Custom filter
- Chaining. {{expression | filter1 | filter2}}
- Arguments. {{expression | filter1:arg1:arg2}}

DEMO 4: FILTERS

\$HTTP SERVICE

Core Angular service that facilitates communication with remote http servers

```
$http.get('/someUrl').  
  success(function(data, status) {  
    //asynchronous callback on success  
  }).  
  error(function(data, status) {  
    //asynchronous callback on error  
  });  
  
$http.post('/someUrl', {msg:'test'}).  
  success(function(data, status) {  
    //asynchronous callback on success  
  }).  
  error(function(data, status) {  
    //asynchronous callback on error  
  });
```

DEPENDENCY INJECTION

Deals with how components get hold of their dependencies

```
angular.module('myApp', [])  
  .controller('MyRecipeController5', function($http){  
    ...  
    ...  
  })
```

DEMO 5: SERVICE, DEPENDENCY INJECTION

FORMS

Provides validation and data binding

```
<form name="myForm">
  Your fav. recipe
  <input type="text" name="myFavRecipe" ng-model="favRecipe" />
  <span class="error" ng-show="myForm.myFavRecipe.$error.required">
    Required
  </span>

  Your phone no.
  <input type="text" name="myPhoneNumber" ng-model="phoneNumber" />
  <span class="error" ng-show="myForm.myPhoneNumber.$error.minlength">
    Minimum length
  </span>
  <span class="error" ng-show="myForm.myPhoneNumber.$error.maxlength">
    Maximum length
  </span>
</form>
```

DEMO 6: FORMS

SERVICE

Reusable business logic independent of views

```
angular.module('myAppService', [])  
  .factory('recipeService', function() {  
    var recipeList = ["Fruit Tart", "Ice Cream", "Froyo",  
    return {recipeNames: recipeList};  
  })
```


DEMO 7: SERVICE

SPA AND ANGULAR UI ROUTER

- Single Page Applications
- UI-Router - routing framework for AngularJS
- ngRoute vs. UI Router

DEMO 8: ANGULAR UI ROUTER

RESOURCES

- Official Angular documentation <https://docs.angularjs.org>
- Angular UI Router <https://github.com/angular-ui/ui-router>
- Unit & E2E Testing in Angular
<https://docs.angularjs.org/guide/unit-testing>
- John Papa's Angular Style Guide
<https://github.com/johnpapa/angular-styleguide>
- Adventures in Angular Podcast
<http://devchat.tv/adventures-in-angular>
- ng-wat talk by Shai Reznik
https://www.youtube.com/watch?v=M_Wp-2XA9ZU

QUESTIONS?