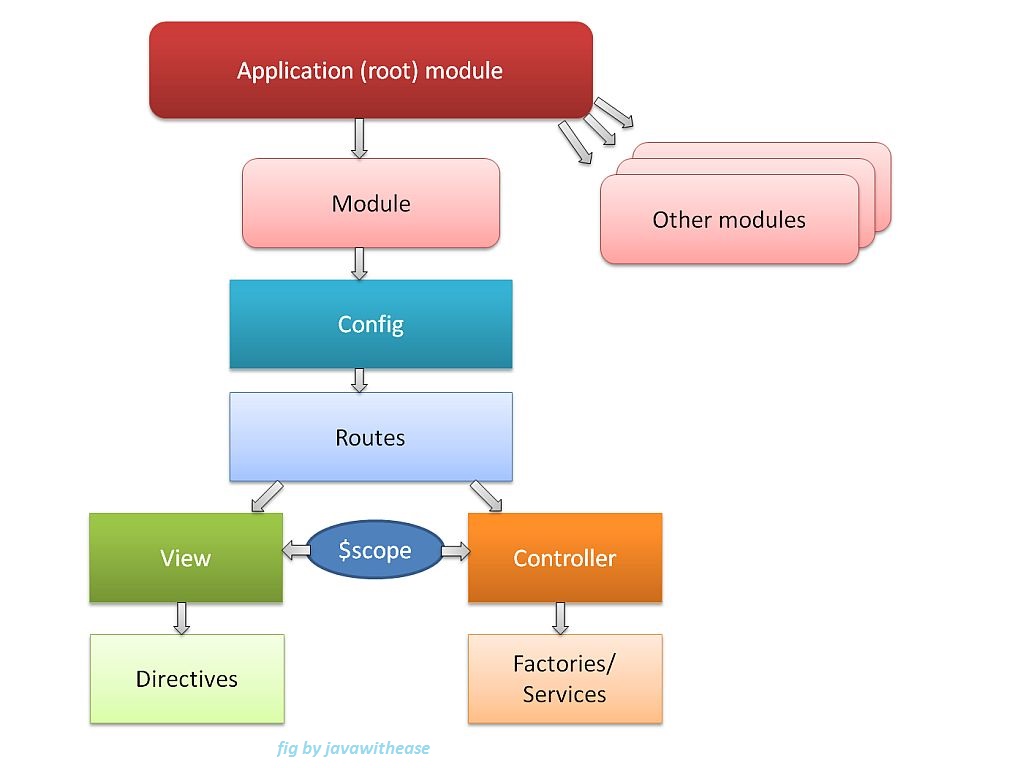
**Angular JS**

Angular is a front-end framework built to ease the burden of writing complex apps while keeping everything testable and organized. JavaScript Frameworks for Building SPAs (“single-page application”).



**Figure 1: AngularJS architecture.**

**AngularJS core features:**

1. Modules

2. Directives

3. Templates

4. Scope

5. Expressions

6. Data-Binding

7. MVC (Model, View & Controller)

8. Validations

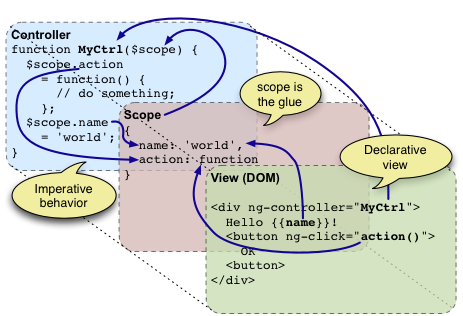
9. Filters

10. Services

11. Routing

12. Dependency Injection

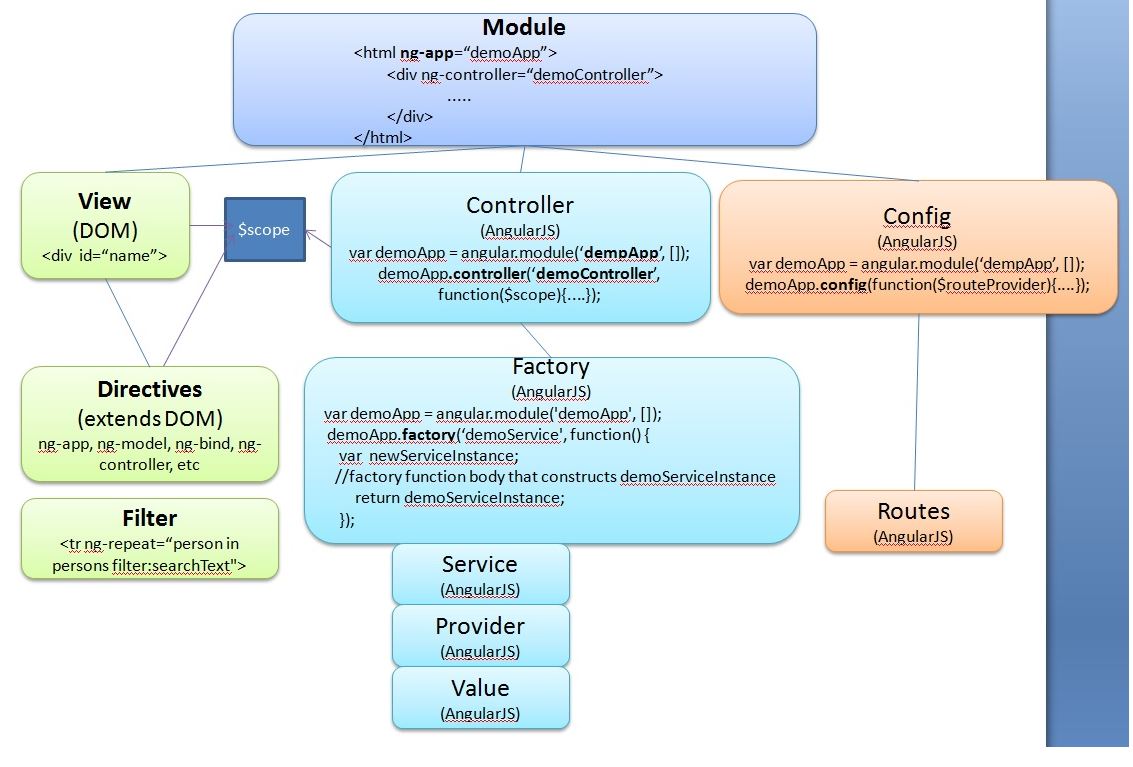
13. Testing.



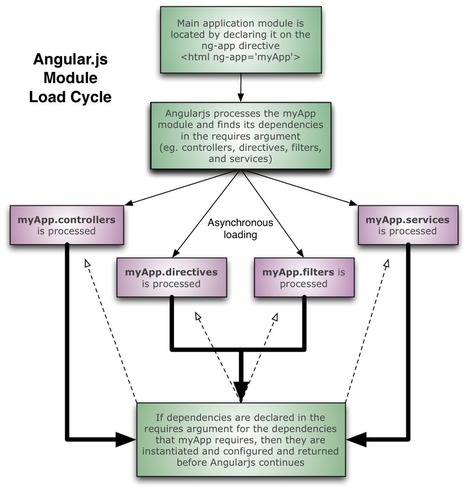
**Figure 2: Two-way data binding.**

AngularJS is a library written in JavaScript and can be included in a web page before the closing body tag.

**<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>**

****

**Figure 3: Angular JS overview.**

****

**Figure 3: Angular JS load cycle.**

**Table 1:** **AngularJS Directives.**

|  |  |
| --- | --- |
| **Directive** | **Description** |
| [ng-app](https://www.w3schools.com/angular/ng_ng-app.asp) | Defines the root element of an application. |
| [ng-bind](https://www.w3schools.com/angular/ng_ng-bind.asp) | Binds the content of an HTML element to application data. |
| [ng-bind-html](https://www.w3schools.com/angular/ng_ng-bind-html.asp) | Binds the innerHTML of an HTML element to application data, and also removes dangerous code from the HTML string. |
| [ng-bind-template](https://www.w3schools.com/angular/ng_ng-bind-template.asp) | Specifies that the text content should be replaced with a template. |
| [ng-blur](https://www.w3schools.com/angular/ng_ng-blur.asp) | Specifies a behavior on blur events. |
| [ng-change](https://www.w3schools.com/angular/ng_ng-change.asp) | Specifies an expression to evaluate when content is being changed by the user. |
| [ng-checked](https://www.w3schools.com/angular/ng_ng-checked.asp) | Specifies if an element is checked or not. |
| [ng-class](https://www.w3schools.com/angular/ng_ng-class.asp) | Specifies CSS classes on HTML elements. |
| [ng-class-even](https://www.w3schools.com/angular/ng_ng-class-even.asp) | Same as ng-class, but will only take effect on even rows. |
| [ng-class-odd](https://www.w3schools.com/angular/ng_ng-class-odd.asp) | Same as ng-class, but will only take effect on odd rows. |
| [ng-click](https://www.w3schools.com/angular/ng_ng-click.asp) | Specifies an expression to evaluate when an element is being clicked. |
| [ng-cloak](https://www.w3schools.com/angular/ng_ng-cloak.asp) | Prevents flickering when your application is being loaded. |
| [ng-controller](https://www.w3schools.com/angular/ng_ng-controller.asp) | Defines the controller object for an application. |
| [ng-copy](https://www.w3schools.com/angular/ng_ng-copy.asp) | Specifies a behavior on copy events. |
| [ng-csp](https://www.w3schools.com/angular/ng_ng-csp.asp) | Changes the content security policy. |
| [ng-cut](https://www.w3schools.com/angular/ng_ng-cut.asp) | Specifies a behavior on cut events. |
| [ng-dblclick](https://www.w3schools.com/angular/ng_ng-dblclick.asp) | Specifies a behavior on double-click events. |
| [ng-disabled](https://www.w3schools.com/angular/ng_ng-disabled.asp) | Specifies if an element is disabled or not. |
| [ng-focus](https://www.w3schools.com/angular/ng_ng-focus.asp) | Specifies a behavior on focus events. |
| ng-form | Specifies an HTML form to inherit controls from. |
| [ng-hide](https://www.w3schools.com/angular/ng_ng-hide.asp) | Hides or shows HTML elements. |
| [ng-href](https://www.w3schools.com/angular/ng_ng-href.asp) | Specifies a url for the <a> element. |
| [ng-if](https://www.w3schools.com/angular/ng_ng-if.asp) | Removes the HTML element if a condition is false. |
| [ng-include](https://www.w3schools.com/angular/ng_ng-include.asp) | Includes HTML in an application. |
| [ng-init](https://www.w3schools.com/angular/ng_ng-init.asp) | Defines initial values for an application. |
| ng-jq | Specifies that the application must use a library, like jQuery. |
| [ng-keydown](https://www.w3schools.com/angular/ng_ng-keydown.asp) | Specifies a behavior on keydown events. |
| [ng-keypress](https://www.w3schools.com/angular/ng_ng-keypress.asp) | Specifies a behavior on keypress events. |
| [ng-keyup](https://www.w3schools.com/angular/ng_ng-keyup.asp) | Specifies a behavior on keyup events. |
| [ng-list](https://www.w3schools.com/angular/ng_ng-list.asp) | Converts text into a list (array). |
| [ng-maxlength](https://www.w3schools.com/angular/ng_ng-maxlength.asp) | Specifies the maximum number of characters allowed in the input field. |
| [ng-minlength](https://www.w3schools.com/angular/ng_ng-minlength.asp) | Specifies the minimum number of characters allowed in the input field. |
| [ng-model](https://www.w3schools.com/angular/ng_ng-model.asp) | Binds the value of HTML controls to application data. |
| [ng-model-options](https://www.w3schools.com/angular/ng_ng-model-options.asp) | Specifies how updates in the model are done. |
| [ng-mousedown](https://www.w3schools.com/angular/ng_ng-mousedown.asp) | Specifies a behavior on mousedown events. |
| [ng-mouseenter](https://www.w3schools.com/angular/ng_ng-mouseenter.asp) | Specifies a behavior on mouseenter events. |
| [ng-mouseleave](https://www.w3schools.com/angular/ng_ng-mouseleave.asp) | Specifies a behavior on mouseleave events. |
| [ng-mousemove](https://www.w3schools.com/angular/ng_ng-mousemove.asp) | Specifies a behavior on mousemove events. |
| [ng-mouseover](https://www.w3schools.com/angular/ng_ng-mouseover.asp) | Specifies a behavior on mouseover events. |
| [ng-mouseup](https://www.w3schools.com/angular/ng_ng-mouseup.asp) | Specifies a behavior on mouseup events. |
| [ng-non-bindable](https://www.w3schools.com/angular/ng_ng-non-bindable.asp) | Specifies that no data binding can happen in this element, or its children. |
| [ng-open](https://www.w3schools.com/angular/ng_ng-open.asp) | Specifies the open attribute of an element. |
| [ng-options](https://www.w3schools.com/angular/ng_ng-options.asp) | Specifies <options> in a <select> list. |
| [ng-paste](https://www.w3schools.com/angular/ng_ng-paste.asp) | Specifies a behavior on paste events. |
| ng-pluralize | Specifies a message to display according to en-us localization rules. |
| [ng-readonly](https://www.w3schools.com/angular/ng_ng-readonly.asp) | Specifies the readonly attribute of an element. |
| [ng-repeat](https://www.w3schools.com/angular/ng_ng-repeat.asp) | Defines a template for each data in a collection. |
| [ng-required](https://www.w3schools.com/angular/ng_ng-required.asp) | Specifies the required attribute of an element. |
| [ng-selected](https://www.w3schools.com/angular/ng_ng-selected.asp) | Specifies the selected attribute of an element. |
| [ng-show](https://www.w3schools.com/angular/ng_ng-show.asp) | Shows or hides HTML elements. |
| [ng-src](https://www.w3schools.com/angular/ng_ng-src.asp) | Specifies the src attribute for the <img> element. |
| [ng-srcset](https://www.w3schools.com/angular/ng_ng-srcset.asp) | Specifies the srcset attribute for the <img> element. |
| [ng-style](https://www.w3schools.com/angular/ng_ng-style.asp) | Specifies the style attribute for an element. |
| [ng-submit](https://www.w3schools.com/angular/ng_ng-submit.asp) | Specifies expressions to run on onsubmit events. |
| [ng-switch](https://www.w3schools.com/angular/ng_ng-switch.asp) | Specifies a condition that will be used to show/hide child elements. |
| ng-transclude | Specifies a point to insert transcluded elements. |
| [ng-value](https://www.w3schools.com/angular/ng_ng-value.asp) | Specifies the value of an input element. |

**Table 2: Angular JS filters.**

|  |  |
| --- | --- |
| **Filter** | **Description** |
| [currency](https://www.w3schools.com/angular/ng_filter_currency.asp) | Format a number to a currency format. |
| [date](https://www.w3schools.com/angular/ng_filter_date.asp) | Format a date to a specified format. |
| [filter](https://www.w3schools.com/angular/ng_filter_filter.asp) | Select a subset of items from an array. |
| [json](https://www.w3schools.com/angular/ng_filter_json.asp) | Format an object to a JSON string. |
| [limitTo](https://www.w3schools.com/angular/ng_filter_limitto.asp) | Limits an array, or a string, into a specified number of elements/characters. |
| [lowercase](https://www.w3schools.com/angular/ng_filter_lowercase.asp) | Format a string to lower case. |
| [number](https://www.w3schools.com/angular/ng_filter_number.asp) | Format a number to a string. |
| [orderBy](https://www.w3schools.com/angular/ng_filter_orderby.asp) | Orders an array by an expression. |
| [uppercase](https://www.w3schools.com/angular/ng_filter_uppercase.asp) | Format a string to upper case. |

**Table 3:** **AngularJS Global API Converting.**

|  |  |
| --- | --- |
| **API** | **Description** |
| angular.lowercase() | Converts a string to lowercase |
| angular.uppercase() | Converts a string to uppercase |
| angular.copy() | Creates a deep copy of an object or an array |
| angular.forEach() | Executes a function for each element in an object or array |

**Table 4:** **AngularJS Global API Comparing.**

|  |  |
| --- | --- |
| **API** | **Description** |
| angular.isArray() | Returns true if the reference is an array |
| angular.isDate() | Returns true if the reference is a date |
| angular.isDefined() | Returns true if the reference is defined |
| angular.isElement() | Returns true if the reference is a DOM element |
| angular.isFunction() | Returns true if the reference is a function |
| angular.isNumber() | Returns true if the reference is a number |
| angular.isObject() | Returns true if the reference is an object |
| angular.isString() | Returns true if the reference is a string |
| angular.isUndefined() | Returns true if the reference is undefined |
| angular.equals() | Returns true if two references are equal |

**Table 5:** **AngularJS Global API JSON.**

|  |  |
| --- | --- |
| **API** | **Description** |
| angular.fromJson() | Takes a JSON string and returns an JavaScript object |
| angular.toJson() | Takes a JavaScript object and returns a JSON string |

**Table 6:** **AngularJS Global API Basics.**

|  |  |
| --- | --- |
| **API** | **Description** |
| angular.bootstrap() | Starts AngularJS manually |
| angular.element() | Wraps an HTML element as an jQuery element |
| angular.module() | Creates, registers, or retrieves an AngularJS module |

**AngularJS Expressions:**

inside double braces: {{ expression }}.

inside a directive: ng-bind="expression".

**AngularJS Modules:**

defines an application, container for the different parts of an application, container for the application controllers.

1. Creating a Module - created by using the AngularJS function angular.module.
2. Adding a Controller - Add a controller to your application, and refer to the controller with the ng-controller directive.
3. Adding a Directive - Addition to the default directives can add default directives to applications.

**AngularJS Directives:**

Create New Directives in addition to all the built-in AngularJS directives, you can create your own directives. New directives are created by using the directive function. To invoke the new directive, make an HTML element with the same tag name as the new directive. When naming a directive, you must use a camel case name, **w3TestDirective**, but when invoking it, you must use - separated name, **w3-test-directive**:

You can invoke a directive by using:

* Element name
* Attribute
* Class
* Comment

You can restrict your directives to only be invoked by some of the methods.

The legal restrict values are:

* E for Element name
* A for Attribute
* C for Class
* M for Comment

By default, the value is EA, meaning that both Element names and attribute names can invoke the directive.

**AngularJS Model:**

Data binding in AngularJS is the synchronization between the model and the view. AngularJS applications usually have a data model. The data model is a collection of data available for the application. The HTML container where the AngularJS application is displayed, is called the view. The view has access to the model, and there are several ways of displaying model data in the view. You can use the ng-bind directive, which will bind the innerHTML of the element to the specified model property:

<p ng-bind="firstname"></p>

<p>First name: {{firstname}}</p>

<input ng-model="firstname">

**Two-way Binding:**

<div ng-app="App1" ng-controller="Ctrl1">  
    Name: <input ng-model="firstname">  
    <h1>{{firstname}}</h1>  
</div>  
  
<script>

var app = angular.module('App1', []);  
app.controller('Ctrl1', function($scope) {  
    $scope.firstname = "John";  
    $scope.lastname = "Doe";  
});

</script>

**AngularJS Controllers:**

AngularJS controllers control the data of AngularJS applications. AngularJS controllers are regular JavaScript Objects. The ng-controller directive defines the application controller.

<div ng-app="App1" ng-controller="Ctrl1">

First Name: <input type="text" ng-model="firstName"><br>

Last Name: <input type="text" ng-model="lastName"><br>

<br>

Full Name: {{firstName + " " + lastName}}

</div>

<script>

var app = angular.module('App1', []);

app.controller('Ctrl1', function($scope) {

$scope.firstName = "John";

$scope.lastName = "Doe";

});

</script>

The ng-controller="myCtrl" attribute is an AngularJS directive, defines a controller. The Ctrl1 function is a JavaScript function. AngularJS will invoke the controller with a $scope object. In AngularJS, $scope is the application object (contains application variables and functions). The controller creates two properties (variables) in the scope (firstName and lastName). The ng-model directives bind the input fields to the controller properties (firstName and lastName).

**AngularJS Scope:**

The scope is the binding part between the HTML (view) and the JavaScript (controller). The scope is an object with the available properties and methods. The scope is available for both the view and the controller.

* View, which is the HTML.
* Model, which is the data available for the current view.
* Controller, which is the JavaScript function that makes/changes/removes/controls the data.

Then the scope is the Model.

The rootScope is available in the entire application. If a variable has the same name in both the current scope and in the rootScope, the application use the one in the current scope.

**AngularJS Filters:**

AngularJS provides filters to transform data: Filters can be added to expressions by using the pipe character |, followed by a filter.

* **currency** Format a number to a currency format.
* date Format a date to a specified format.
* filter Select a subset of items from an array.
* json Format an object to a JSON string.
* limitTo Limits an array/string, into a specified number of elements/characters.
* lowercase Format a string to lower case.
* number Format a number to a string.
* orderBy Orders an array by an expression.
* uppercase Format a string to upper case.

**Custom Filter:**

app.filter('Format1', **function**() {

**return** **function**(x) {

logic goes here

**return** txt;//format to return

};

});

**AngularJS Tables:**

<table>

<tr ng-repeat="x in names">

<td>{{ x.Name }}</td>

<td>{{ x.Country }}</td>

</tr>

</table>

**AngularJS Select Boxes:**

Dropdowns made with ng-options allows the selected value to be an object, while dropdowns made from ng-repeat has to be a string.

1. <select ng-model="selectedName" ng-options="x for x in names"></select>
2. <select><option ng-repeat="x in names">{{x}}</option></select>

Object with key-value pairs the selected value will always be the value in a key-value pair.The value in a key-value pair can also be an object:

<select ng-model="selectedCar" ng-options="x for (x, y) in cars"></select>

**AngularJS Validation:**

Input fields have the following states:

* $untouched The field has not been touched yet
* $touched The field has been touched
* $pristine The field has not been modified yet
* $dirty The field has been modified
* $invalid The field content is not valid
* $valid The field content is valid

Forms have the following states:

* $pristine No fields have been modified yet
* $dirty One or more have been modified
* $invalid The form content is not valid
* $valid The form content is valid
* $submitted The form is submitted

The following classes are added to, or removed from, input fields:

* ng-untouched The field has not been touched yet
* ng-touched The field has been touched
* ng-pristine The field has not been modified yet
* ng-dirty The field has been modified
* ng-valid The field content is valid
* ng-invalid The field content is not valid
* ng-valid-key One key for each validation. Example: ng-valid-required, useful when there are more than one thing that must be validated
* ng-invalid-key Example: ng-invalid-required

The following classes are added to, or removed from, forms:

* ng-pristine No fields has not been modified yet
* ng-dirty One or more fields has been modified
* ng-valid The form content is valid
* ng-invalid The form content is not valid
* ng-valid-key One key for each validation. Example: ng-valid-required, useful when there are more than one thing that must be validated
* ng-invalid-key Example: ng-invalid-required

Add styles for these classes to give your application a better and more intuitive user interface.

**AngularJS Events:**

You can add AngularJS event listeners to HTML elements by using one or more of these directives:

* ng-blur
* ng-change
* ng-click
* ng-copy
* ng-cut
* ng-dblclick
* ng-focus
* ng-keydown
* ng-keypress
* ng-keyup
* ng-mousedown
* ng-mouseenter
* ng-mouseleave
* ng-mousemove
* ng-mouseover
* ng-mouseup
* ng-paste