





AngularJS is super cool.

By Max.

# AngularJS is MVW.

## **Which is:**

- Model
- View
- WTF? — Whatever

# Whatever?

*— I'd rather see developers build kick-ass apps that are well-designed and follow separation of concerns, than see them waste time arguing about MV\* nonsense. And for this reason, I hereby declare AngularJS to be MVW framework — Model-View-Whatever. Where Whatever stands for “whatever works for you”.*

**Igor Minar**

Source: <https://plus.google.com/+AngularJS/posts/aZNVhj355G2>



# Core concepts

# Core concepts

---

<b>Template</b>	HTML and ...
-----------------	--------------

---

<b>Directive</b>	... a few new attributes*.
------------------	----------------------------

\* With a powerful backend.

---

<b>Model</b>	Just data, nothing more.
--------------	--------------------------

---

<b>Scope</b>	A context where the model is used.
--------------	------------------------------------

The center of the Angular Universe.

---

<b>Expressions</b>	Access to data and functions in the scope.
--------------------	--

---

...

# Core concepts

---

...

---

<b>Filter</b>	Formatting and filtering.
---------------	---------------------------

---

<b>Controller</b>	Business logic, related to View layer.
-------------------	--

---

<b>Service</b>	Reusable business logic, not related with View layer.
----------------	---


---

<b>Module</b>	Container for all this stuff.
---------------	-------------------------------

---

Find out more in the [Conceptual Overview](#).



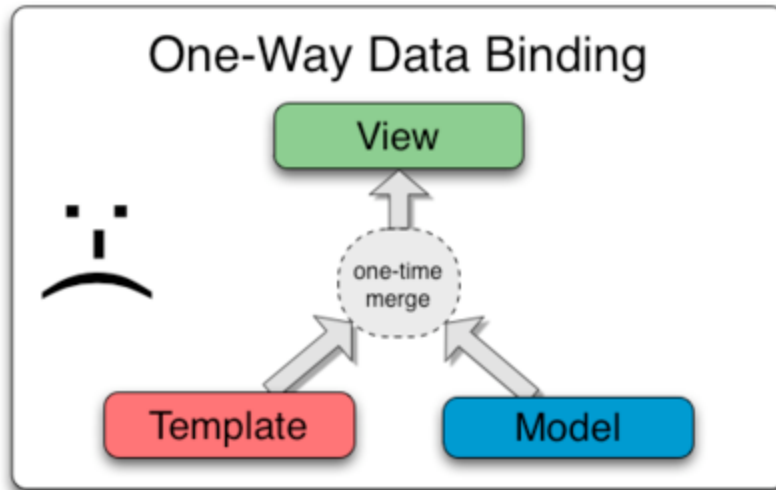


Two-way binding

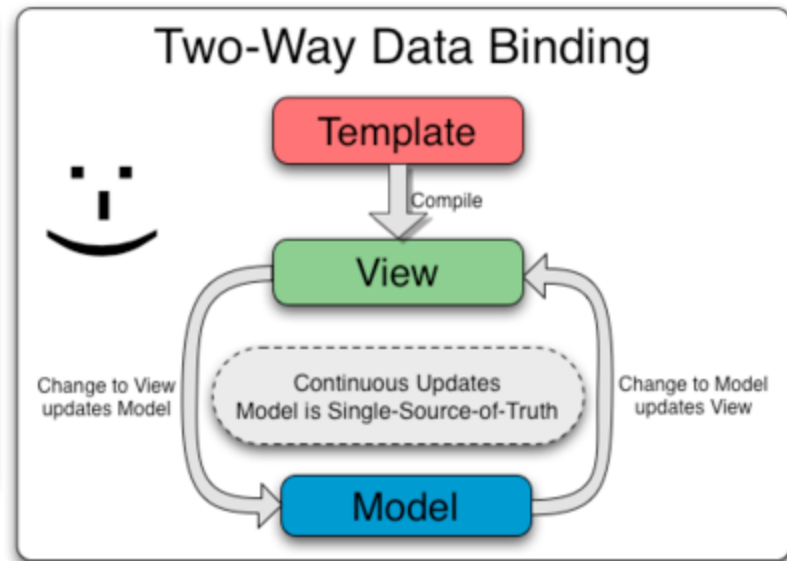


# Data Binding ...

... in Classical Template Systems



... in Angular Templates



# Two-way binding example (supper-dummy)

Enter something:

You've entered:

```
Ø1. <div data-ng-app>
Ø2.   <label>Enter something:</label>
Ø3.   <input type="text" data-ng-model="something" />
Ø4.   <p>You've entered: <strong>{{ something }}</strong></p>
Ø5. </div>
```

— No JavaScript. At all.



# Two-way binding example <sup>(pro)</sup> (still so dummy)

Add contact:

add

All contacts:

[]

---

Filter by:

Filtered:

# Two-way binding example <sup>(pro)</sup> (still so dummy)

Add contact:

All contacts:

[]

```
01. <div data-ng-controller="example2Controller">
02.   <div>
03.     <label>Add contact:<br>
04.       <input type="text" data-ng-model="newContact" />
05.     </label>
06.     <button data-ng-click="addContact(newContact)">add</button>
07.     <div data-ng-show="displayHelp">
08.       <small>Format is wrong</small>
09.     </div>
10.   </div>
11.   <div>
12.     All contacts:<br><small>{{ contacts | json }}</small>
13.   </div>
14.   <!-- ... -->
15. </div>
```



# Two-way binding example <sup>(pro)<sup>(still so dummy)</sup></sup>

Add contact:

All contacts:

[]

```
01. angular.module("angulardemo")
02.   .controller("example2Controller", function($scope) {
03.     $scope.contacts = [];
04.     $scope.addContact = function(newContact) {
05.       var contact = parseContact(newContact);
06.       if (!contact) {
07.         $scope.displayHelp = true;
08.         return;
09.       }
10.       $scope.contacts.push(contact);
11.       $scope.newContact = "";
12.       $scope.displayHelp = false;
13.     };
14.     // ...
15.   });
```

# Two-way binding example <sup>(pro)<sup>(still so dummy)</sup></sup>

Add contact:

All contacts:

[]

```
01. angular.module("angulardemo")
02.   .controller("example2Controller", function($scope) {
03.     // ...
04.
05.     $scope.$watch("newContact", function(newValue, oldValue) {
06.       if (!newValue.length) {
07.         $scope.displayHelp = false;
08.       }
09.     });
10.   });
```

# Two-way binding example <sup>(pro) (still so dummy)</sup>

Add contact:

add

Filter by:

Filtered:

```
01. <input type="text" data-ng-model="query" />
02. <!-- ... -->
03. <ul><li data-ng-repeat="contact in contacts | filter:query">
04.     <small>{{ contact | contactInfo }}</small>
05. </li></ul>
```

```
01. angular.module("angulardemo")
02.     .controller("example2Controller", function($scope) { /* ... */ });
03.     .filter("contactInfo", function() {
04.         return function(input) {
05.             var name = input.firstName && input.lastName
06.                 ? input.firstName + " " + input.lastName : null;
07.             return input.email
08.                 ? (name ? name + " <" + input.email + ">" : input.email) : name;
09.         }
10.     });
```



# Dependency injection

# Dependency injection example <sup>(dummy)</sup>

UAH:

**USD:** 0.00    **EUR:** 0.00    **GBP:** 0.00

```
01. <div>UAH: <input type="text" data-ng-model="amount" /></div>
02. <div>USD: {{ convert(amount, "UAH", "USD") | number:2 }}</div>
03. <div>EUR: {{ convert(amount, "UAH", "EUR") | number:2 }}</div>
04. <div>GBP: {{ convert(amount, "UAH", "GBP") | number:2 }}</div>
```

```
01. angular.module("angulardemo")
02.   .controller("example3Controller", function($scope, ExchangeRateDummy) {
03.     $scope.convert = function(amount, from, to) {
04.       var rate = ExchangeRateDummy.getRate(from, to);
05.       return (parseFloat(amount) || 0) * (rate.rate || 0);
06.     }
07.   });
```



# Dependency injection example <sup>(dummy)</sup>

UAH:

**USD: 0.00   EUR: 0.00   GBP: 0.00**

```
01. angular.module("angulardemo")
02.   .factory("ExchangeRateDummy", function() {
03.     var rates = {
04.       "UAHUSD": { from: "UAH", to: "USD", rate: 0.06 },
05.       "UAHEUR": { from: "UAH", to: "EUR", rate: 0.05 },
06.       "UAHGBP": { from: "UAH", to: "GBP", rate: 0.04 }
07.     };
08.     return {
09.       getRate: function(from, to) {
10.         return (from + to in rates) ? rates[from + to] : {};
11.       }
12.     }
13.   });
```

# Dependency injection example <sup>(pro)</sup>

**USD:**

**AUD:** 0.00   **EUR:** 0.00   **GBP:** 0.00

```
Ø1. <div>USD: <input type="text" data-ng-model="amount" /></div>
Ø2. <div>AUD: {{ converted.AUD | number:2 }}</div>
Ø3. <div>EUR: {{ converted.EUR | number:2 }}</div>
Ø4. <div>GBP: {{ converted.GBP | number:2 }}</div>
```

# Dependency injection example <sup>(pro)</sup>

**USD:**

**AUD:** 0.00 **EUR:** 0.00 **GBP:** 0.00

```
01. angular.module("angulardemo")
02.   .controller("example4Controller", function($scope, ExchangeRate) {
03.     $scope.converted = {"AUD": 0, "EUR": 0, "GBP": 0};
04.     $scope.$watch("amount", function(amount) {
05.       angular.forEach($scope.converted, function(convertedValue, currency) {
06.         ExchangeRate.getRate("USD", currency).then(function(rate) {
07.           $scope.converted[currency] = (parseFloat(amount) || 0) * (rate || 0);
08.         });
09.       });
10.     });
11.   })
12.   .factory("ExchangeRate", ["$http", function($http) {
13.     var apiUrl = "http://apilayer.net/api/live?access_key=XXX";
14.     return {
15.       getRate: function(from, to) {
16.         var requestUrl = apiUrl + "&source=" + from + "&currencies=" + to
17.           + "&callback=JSON_CALLBACK";
18.         return $http.jsonp(requestUrl, {cache: true}).then(function(response) {
19.           return response.data.quotes[from + to];
20.         });
21.       }
22.     }
23.   });
```

# SCIENCE!





# Directives

in a [real-life project](#)



# Cache manager directive

```
Ø1. <!-- app/templates/index.html -->
Ø2. <footer class="container-fluid">
Ø3.     ...
Ø4.     <div data-cache-manager></div>
Ø5.     ...
Ø6. </footer>
```

# Cache manager directive

```
01. // app/js/modules/agile/directives/cache_manager.js
02. angular.module("agile.directives")
03.     .directive("cacheManager",
04.         function($templateCache, Storage, Helper, TEMPLATES_URL) {
05.             return {
06.                 scope: {},
07.                 controller: function ($scope) {
08.                     $scope.clear = function() {
09.                         $templateCache.removeAll();
10.                         Storage.clearAll();
11.                         Helper.setAlert("success", "Cache has been cleared.")
12.                     }
13.                 },
14.                 templateUrl: TEMPLATES_URL + "/cache-manager.directive.html"
15.             };
16.         });
```

```
01. <!-- app/templates/cache-manager.directive.html -->
02. <div class="cache-manager pull-right">
03.     <button type="button" data-ng-click="clear()">Clear Cache</button>
04. </div>
```

# Enough code

Just a few more things  
in a conclusion.



# Angular Projects

- <https://builtwith.angularjs.org/>
- <https://github.com/angular/angular.js/wiki/Projects-using-AngularJS>

# Angular Projects

- [Kontaktkarte](#) — Contact book
- [Hatjitsu](#) — Distributed online scrum planning poker
- [Semantic Body Browser](#) — A tool for graphically exploring an organism
- [Drugsearch](#) — Android app for searching drugs in dragstores
- [Mapa de personajes de la serie Isabel](#) — The characters map in the “Isabel” series; ♥



# Angular is for:

- **Web apps** (single-page apps, mobile apps, browser extensions, etc.)
- **Interactive websites** (social networks, content management, admin panels)
- **Interactive pages** (checkout, statistics, graphics, reports, etc.)
- **Fun**

# Angular is not for:

- Blogs
- News
- Affiliates
- Corporate websites
- E-Commerce
- Wikis

Of course it's not a rule.