

Learn to Code Melbourne: Foundations of AngularJS

The Future of Web Development

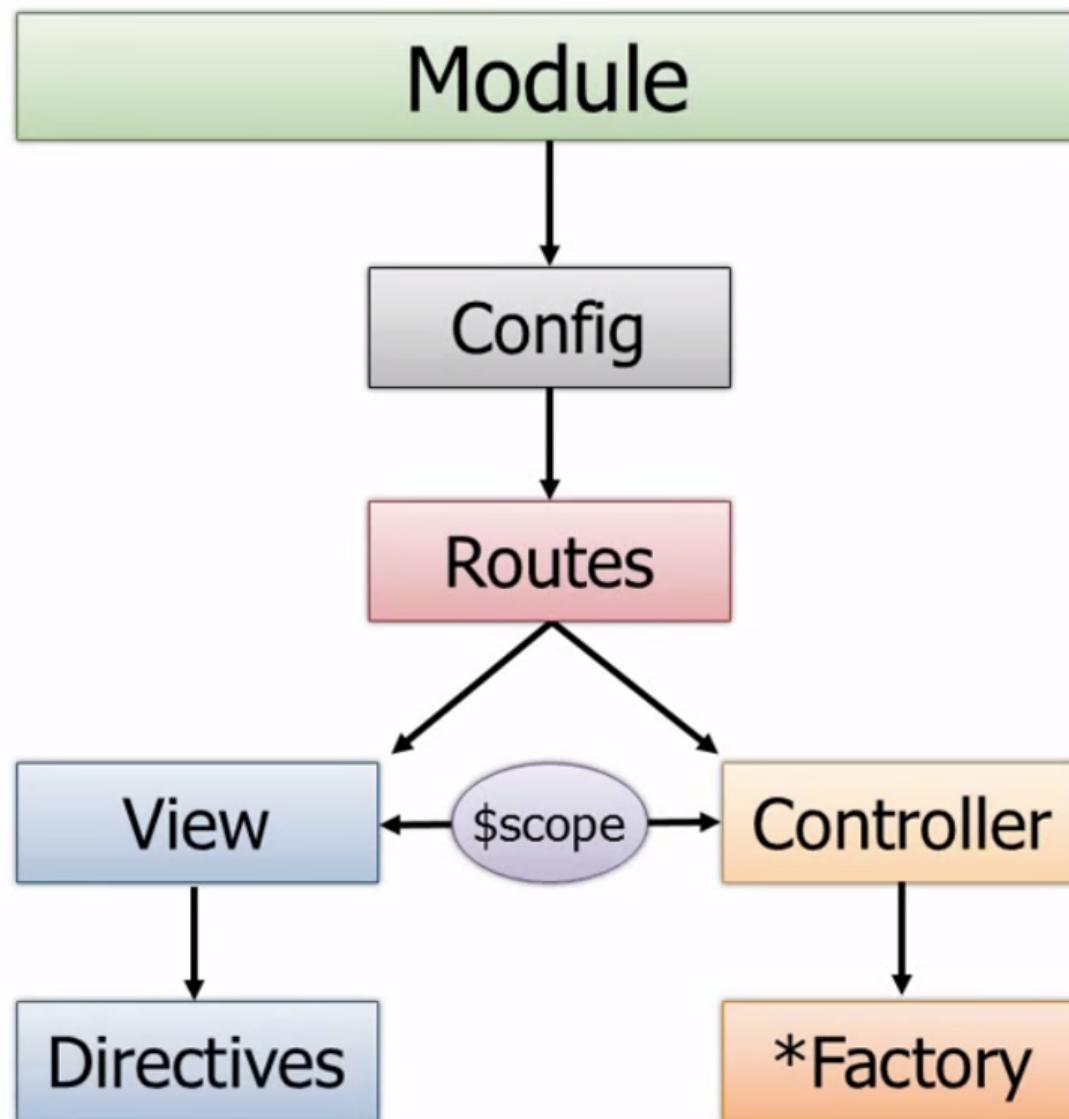
Goals for Today

- Learn the basics of Angular
(ng-App, ng-Controller, ng-Model)
- Understanding Angular Scope
- Directives, Filters, & Services

Goals for Tomorrow

- Learn the new architecture of AngularJS 1.5
- Migration strategies from AngularJS 1.x to 2.0

The AngularJS Structure



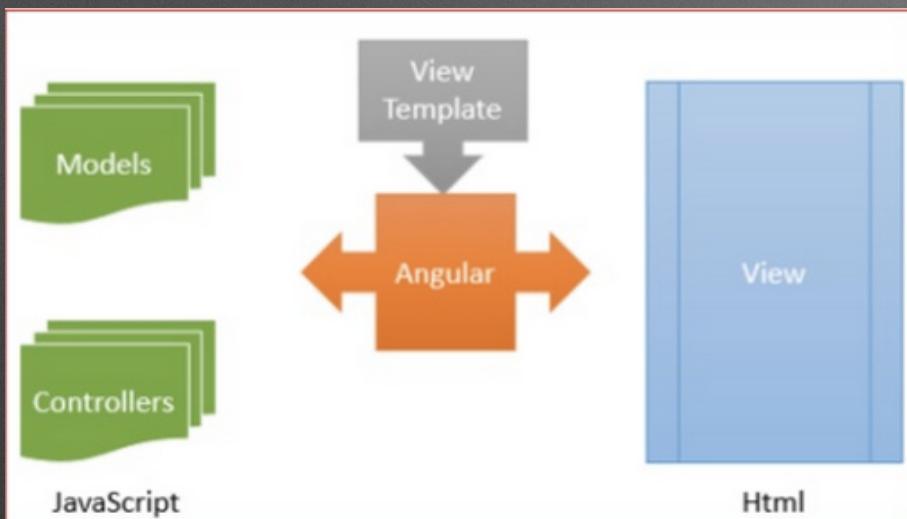
Design Patterns

- Model-View-Controller
- Model-View-ViewModel
- Modules: `Angular.module()`
- Dependency Injection
`Angular.module([dependencies])`

Design Patterns

- Pub/Sub: \$broadcast & \$emit
- Singletons: angular.Service
- Observer: \$watch

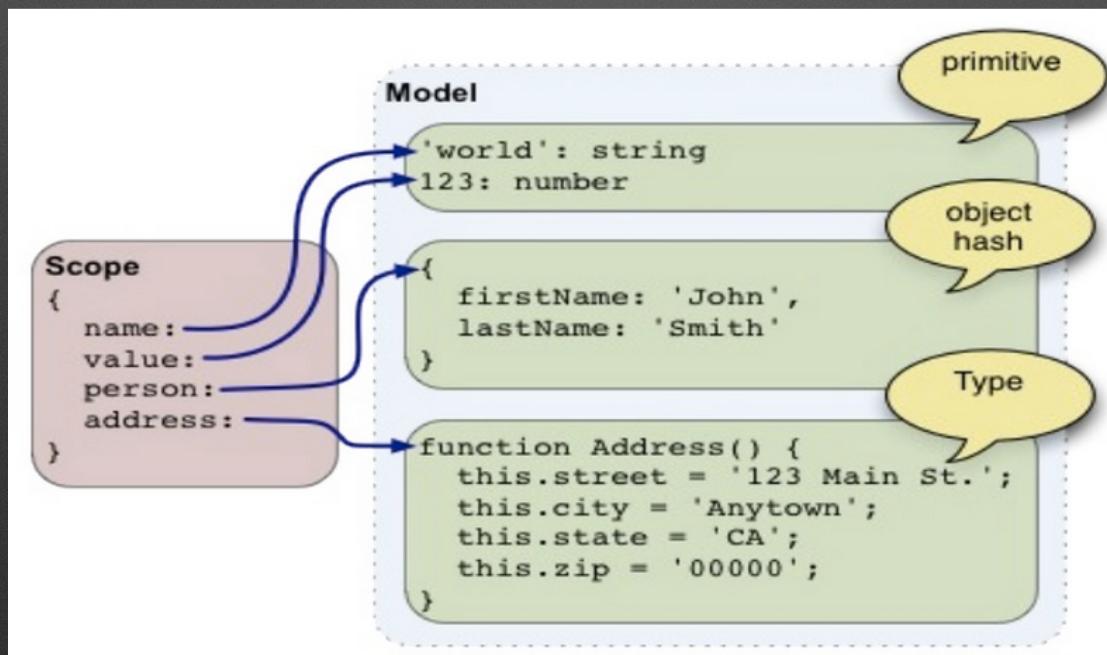
Model View Controller



- Model + View + ViewModel
- Model: Javascript Variables
- View: ng-model
- ViewModel: Controllers & \$scope

Model

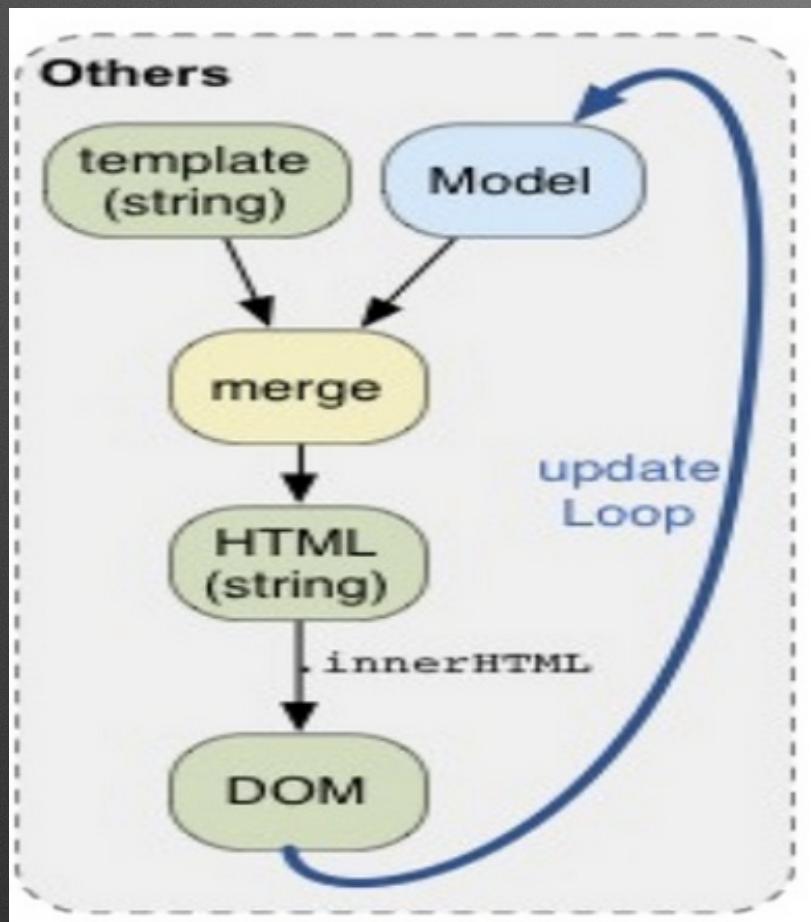
- Model is a javascript object.
- \$scope



Controller

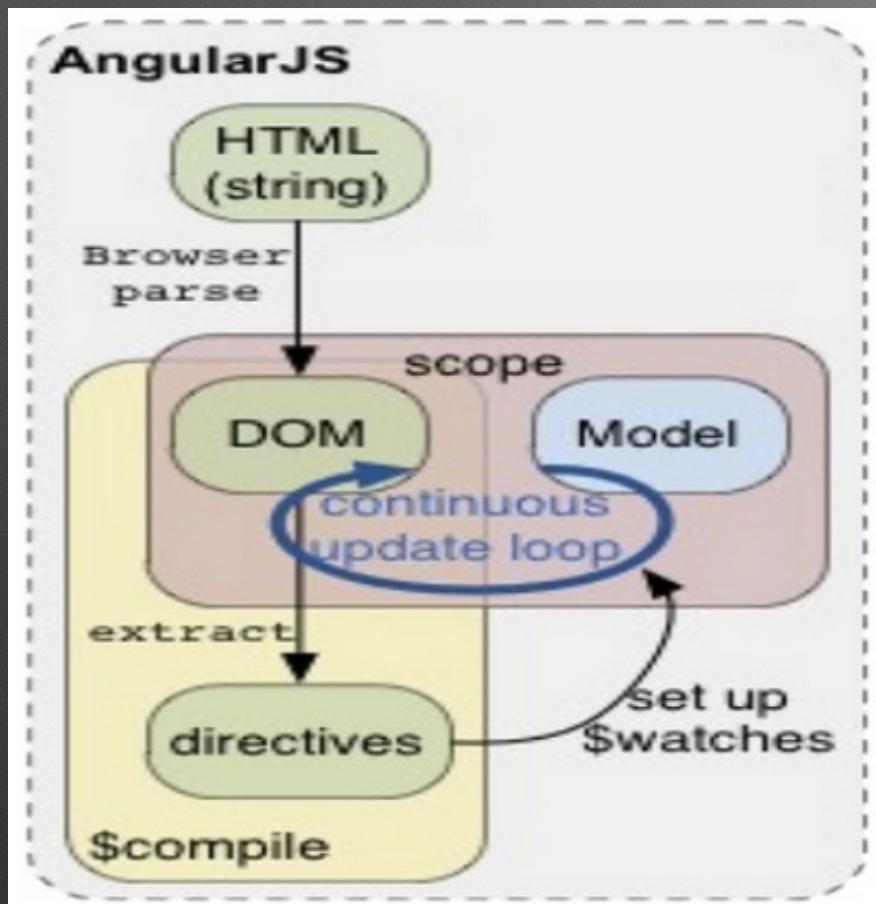
- Controller is the implementation of code
- Creates the Model (ie, \$scope), and sends it to the View
- A Controller's functions can be set as functions in the DOM
 - Remember onClick="alert()" ?

jQuery Updates



1. Template + Model = HTML String
2. When DOM changes, we manually call an Update Loop
3. Update Loop updates the Model

Angular Updates



1. A continuous update loop checks if the DOM has changed
2. When DOM changes, Angular updates the Model automatically
3. We can also watch for updates

List of Angular Built-in Directives 1/2

- **ng-app**
Declares a HTML element to be the root of the NG application
- **ng-controller**
An encapsulation of logic specific to a view
- **ng-bind**
Changes the .text() of a HTML element (one way binding)
- **ng-model**
Declares a two way binding between the HTML element and Javascript

List of Angular Built-in Directives 2/2

- **ng-repeat**
Expands a javascript array into individual HTML elements
- **ng-if**
Writes HTML to the DOM if true,
- **ng-show/ng-hide**
Applies a ng-hide, or ng-show class to the HTML element.
- **ng-view**
Placeholder for NG routes to insert HTML templates

Angular Walkthrough 1/3

- **ng-app**
Declares a HTML element to be the root of the NG application
- **ng-controller**
An encapsulation of logic specific to a view
- **\$scope**
The JS object that binds with the HTML DOM

Starting an AngularJS Project

1. Import Angular.js file
2. Add ng-app tag to the HTML page
`<html ng-app>`
or `<html ng-app="myApp">`
or `<div ng-app="myApp">`
3. Add a controller
`<div ng-controller="PageController">`
4. Implement the Controller
`function PageController($scope) { }`

Scope

- \$scope is found within a Controller
- \$rootScope is at the ng-app level
- Chrome Console
`angular.element($0).scope()`

Angular Walkthrough 2/3

- **ng-bind**
Changes the `.text()` of a HTML element (one way binding)
Also use `{{Model.Name}}`
- **ng-model**
Declares a two way binding between the HTML element and Javascript.
When do we need two way binding?

Angular Walkthrough 3/3

- **ng-repeat**
Expands a javascript array into individual HTML elements
- **ng-if**
Writes HTML to the DOM if true,
- **ng-show/ng-hide**
Applies a ng-hide, or ng-show class to the HTML element.
- **ng-class**
Applies a CSS class if a statement == true

Let's work on a Project...

1. git clone <https://bit.ly/ngtutorial>
or
`git clone https://bitbucket.org/tiang/ngtutorial.git`
2. cd ngtutorial
3. npm install

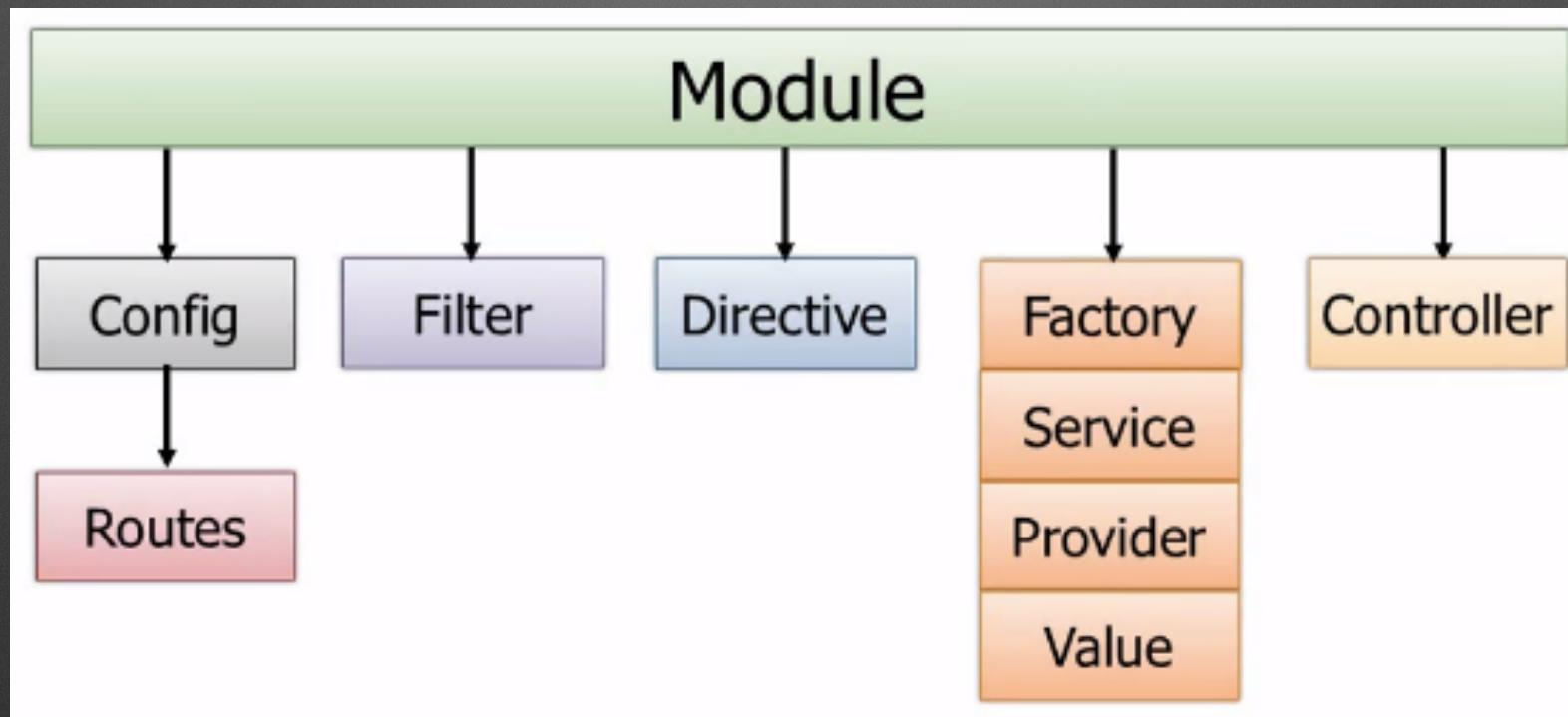
AngularJS - The Next Level

The Future of Web Development (I Promise!)

Helpers

- Modules
- Filters
- Routes
- Directives
- Services

The AngularJS Structure



Modules - Container of Code

Creating a Module

What's the
Array for?

```
var demoApp = angular.module('demoApp', []);
```

```
var demoApp = angular.module('demoApp',  
['helperModule']);
```

Module that demoApp
depends on

Filters

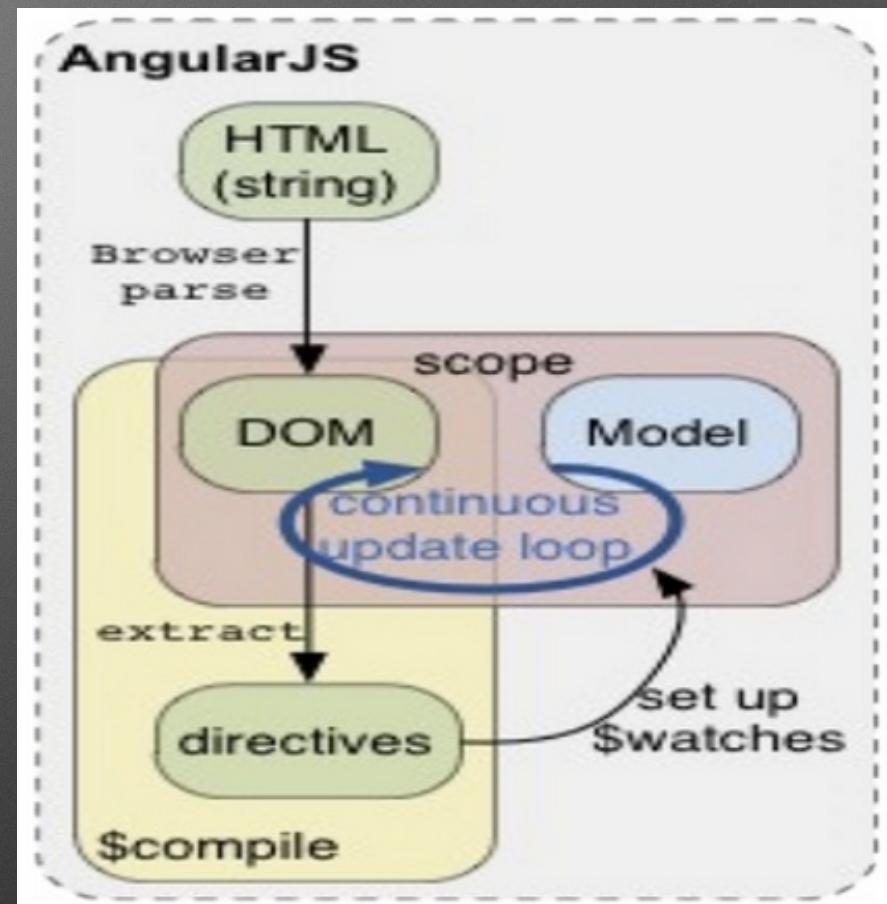
- Simple and quick way to modify your Model
- Modifies the Model during the continuous update loop

Using filters in view templates

Filters can be applied to expressions in view templates using the following syntax:

```
 {{ expression | filter }}
```

E.g. the markup `{{ 12 | currency }}` formats the number 12 as a currency using the `currency` filter. The resulting value is `$12.00`.



Routes

- Angular Routes load Controllers and HTML templates
- Also pre-loads data and dependencies
- Has events you can attach functions to

Routes

- Has events you can attach functions to
- \$routeChangeSuccess
- \$routeChangeStart

```
$rootScope.$on('$routeChangeSuccess', function() {  
  batchLog($route.current ? $route.current.template : null);  
});
```

Directives

- Tells the Angular Framework to compile the HTML
- Reusable JS/HTML that you can apply in the DOM
- We have already been using Directives
ng-app, ng-model
- Apply code functionality to specific DOM via
“Compilation” (Recursive)

Directives

What does it mean to "compile" an HTML template? For AngularJS, "compilation" means attaching event listeners to the HTML to make it interactive. The reason we use the term "compile" is that the recursive process of attaching directives mirrors the process of compiling source code in compiled programming languages.

Directives

- “Normalisation”: The process of matching the Directive in the DOM, with the Directive in Javascript
- 1. Strip x- and data- from the front of the element/attributes.
- 2. Convert the :, -, or _-delimited name to camelCase.
- ng-bind => directive('ngBind', function() { ... })

Directives

- directive('nameOfDirective', function() {
 return {
 templateUrl: 'filename',
 link: function() { ... },
 restrict: 'A, E, C',
 scope: { 'name', '=', '&', '@' }
 transclude: true, false,
 require: '^ControllerName'
 }
})

Directives

- Lets make one of the forms into a Directive

FireBase.io

- Database as a Service
- Provides real-time sockets streaming of data
- Data: `Firebase.Set({ })`
- Arrays: `Firebase.Push ([{....}, {}, {}])`

FireBase.io

- Database as a Service
- Provides real-time sockets streaming of data
- Data: `Firebase.Set({ })`
- Arrays: `Firebase.Push ([{....}, {}, {}])`

FireBase.io - Creating a Table Row

1. URL = “firebase.io” + “table name”
2. To Create new TableRow:
row = URL.child(‘table row name’)
3. row.set(data);

```
/* We will push the data to the Firebase Database */
/* Key, Value */
$scope.FirbaseDonors = new Firebase('https://blistering-fire-4774.firebaseio.com/donors');
var newtableRow = $scope.FirbaseDonors.child($scope.donorForm.Name);
newtableRow.set(
  $scope.donorForm
);
```

FireBase.io - Reading a Row

1. **tableRow = “firebase.io/table name/table row name”**
2. **tableRow.on(‘value’ function(data) {
 TableRowData = data
});**

```
$scope.getDonorDetails = function(name) {  
  var usersRef = new Firebase('https://blistering-fire-4774.firebaseio.com/donors/'+ name);  
  usersRef.on('value', function(data) {  
    console.log(data.val());  
  });  
}
```

Services/Factory

- Singleton (only one Instance in Memory)
- Examples of only one thing in Web Application?
- Only runs when it is called, and not preloaded.

Bonus Stuff

- Bootstrap.UI - Modals
- Modals Factory
- NG Google Maps (bit.ly/ngGoogleMaps)

Thank you!

Leave us a review on the [meetup.com](https://www.meetup.com) Group Review section if you had a great workshop!