

David Oddoye

- Computer Engineering from KNUST
- Full-Platform-All-Stack, Web and Mobile Developer
- .NET Developer
- Worked for SOFTtribe.

twitter.com/theRealBra_Zee

www.github.com/oddoye-david

oddoyedavid@gmail.com

ANGULARJS

HTML enhanced for web apps!

What is ANGULARJS?

- It's not a JavaScript library (As they say). There are no functions which we can directly call and use.
- It is not a DOM manipulation library like jQuery. But it uses subset of jQuery for DOM manipulation (called jqLite).
- For MVC/MVVM design pattern
- AngularJS is a Javascript MVC framework created by Google to build properly architected and maintainable web applications.



HTML enhanced for web apps!

Philosophy

“ANGULARJS is what HTML would have been if it had been designed for web application development.”

“ANGULARJS is built around the philosophy that declarative code is better than imperative code while building UIs and wiring different components of web application together.”

```
<!doctype html>
<html ng-app>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js"></script>
  </head>
  <body>
    <div>
      <label>Name:</label>
      <input type="text" ng-model="yourName" placeholder="Enter a name here">
      <hr>
      <h1>Hello {{yourName}}!</h1>
    </div>
  </body>
</html>
```

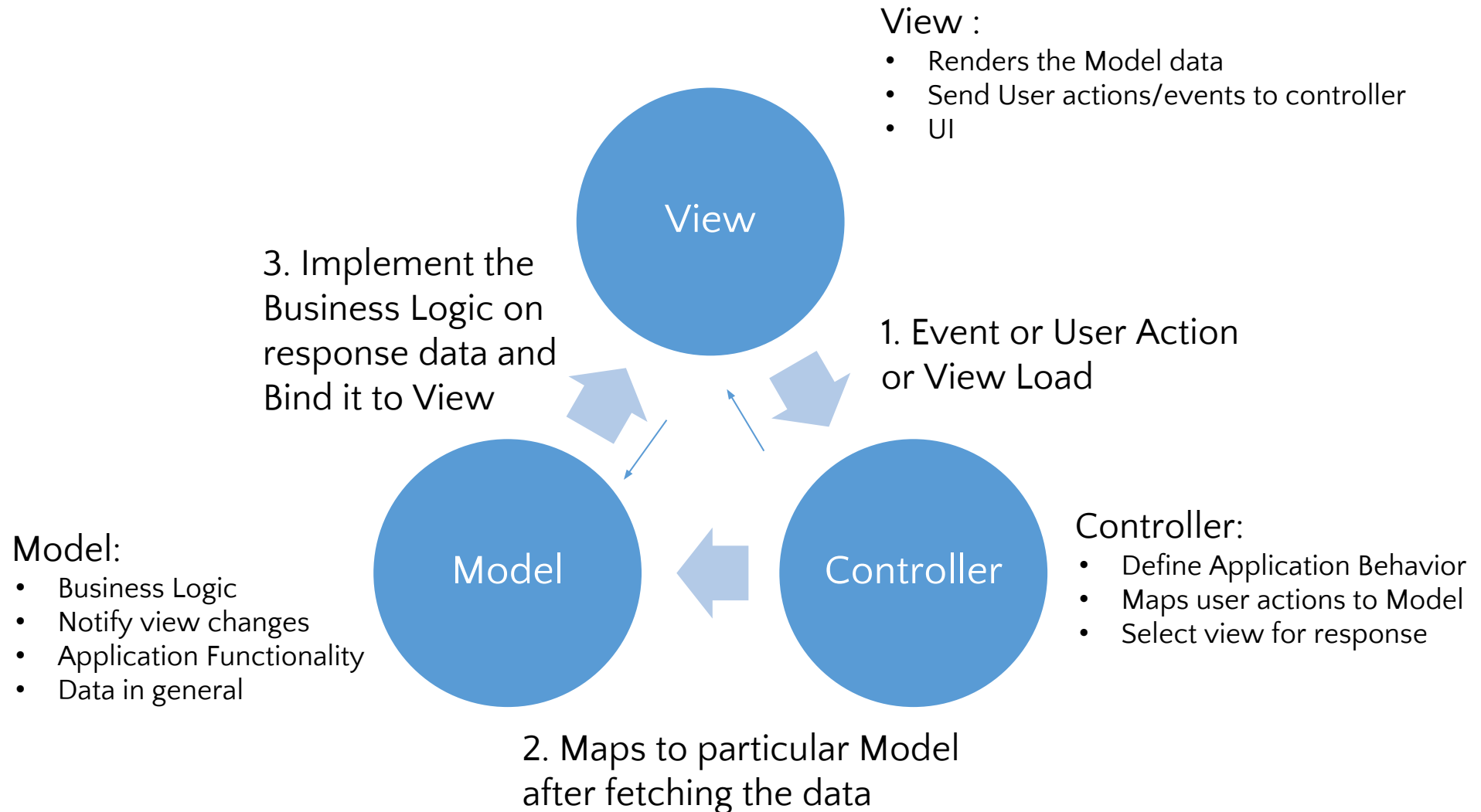
Why ANGULARJS?

- Defines numerous ways to organize web application at client side.
- Enhances HTML by attaching directives, custom tags, attributes, expressions, templates within HTML.
- Encourage TDD
- Encourage MVC/MVVM design pattern
- Code Reuse
- Good for Single Page Apps (SPA)
- Cool Features -> Next Slide

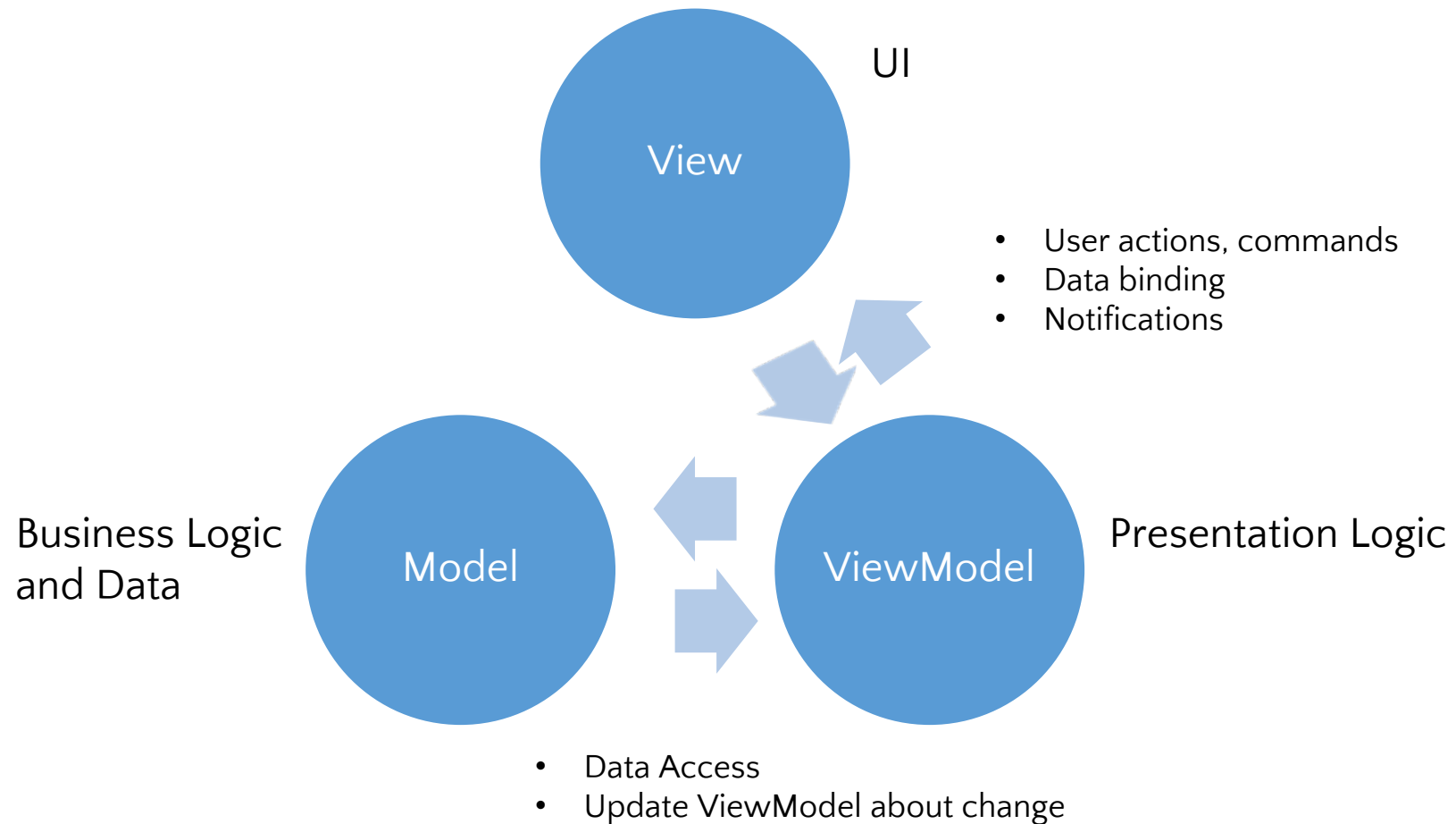
Key Features of ANGULARJS

- Declarative HTML approach
- Easy Data Binding : Two way Data Binding
- Reusable Components
- MVC/MVVM Design Pattern
- Dependency Injection
- End to end Integration Testing / Unit Testing
- Routing
- Templating
- Modules
- Services
- Expressions
- Filters
- Directives
- Form Validation
- `$scope`, `$http`, `$routeProvider`...

MVC : Model View Controller



MVVM: Model View ViewModel



HTML Compiler

Angular's HTML compiler allows the developer to teach the browser new HTML syntax. The compiler allows you to attach behavior to any HTML element or attribute and even create new HTML elements or attributes with custom behavior. Angular calls these behavior extensions directives.

Compiler is an angular service which traverses the DOM looking for attributes. The compilation process happens in two phases.

Compile: traverse the DOM and collect all of the directives. The result is a linking function.

Link: combine the directives with a scope and produce a live view. Any changes in the scope model are reflected in the view, and any user interactions with the view are reflected in the scope model. This makes the scope model the single source of truth.

ng-app

Use this directive to auto-bootstrap an application.

Only one ng-app directive can be used per HTML document

```
<html ng-app>
```

Expression

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

<body>

1+2={{1+2}}

</body>

Directive

The directives can be placed in element names, attributes, class names, as well as comments. Directives are a way to teach HTML new tricks.

A directive is just a function which executes when the compiler encounters it in the DOM.

```
<input ng-model='name'>
```

Custom Defined Directives

```
<span draggable>Drag ME</span>
```

In-Built Directives

- ng-app
- ng-controller
- ng-model
- ng-if
- ng-view
- ng-min
- ng-max
- ng-bind
- ng-class
- ng-style
- ng-include
- ng-show
- ng-hide

Forms

Form and controls provide validation services, so that the user can be notified of invalid input. This provides a better user experience, because the user gets instant feedback on how to correct the error.

```
<input type="text" ng-model="user.name" name="uName" required />
```

```
<button ng-click="update(user)" ng-disabled="form.$invalid || isUnchanged(user)"  
>SAVE</button>
```

Module

Modules declaratively specify how an application should be bootstrapped.

There can be multiple modules in an app

Those could be interdependent too.

```
// declare a module
```

```
var myAppModule = angular.module('myApp', [--here goes the dependent Modules--]);
```

Modules are configured with routes, controllers, models etc.

Scope

Scope is an object that refers to the application model.

It is an execution context for expressions.

Scopes are arranged in hierarchical structure which mimic the DOM structure of the application.

Scopes can watch expressions and propagate events.

Actually the ViewModel of MVVM.

\$scope

Dependency Injection

Dependency Injection (DI) is a software design pattern that deals with how code gets hold of its dependencies.

Filters

Angular filters format data for display to the user.

```
{{ expression [| filter_name[:parameter_value] ... ] }}
```

```
{{ uppercase_expression | uppercase }}
```

```
{{ expression | filter1 | filter2 }}
```

Can create custom filters

Code Time

Fire up them code editors...

Resources

Documentation

- [AngularJS Developer Guide](#)
- [AngularJS API](#)
- [AngularJS Tutorial](#)

Videos

- [AngularJS Fundamentals In 60-ish Minutes](#)
- [Introduction to Angular JS](#)
- [AngularJS end-to-end web app tutorial Part I](#)