YOUSTART

AngularJS

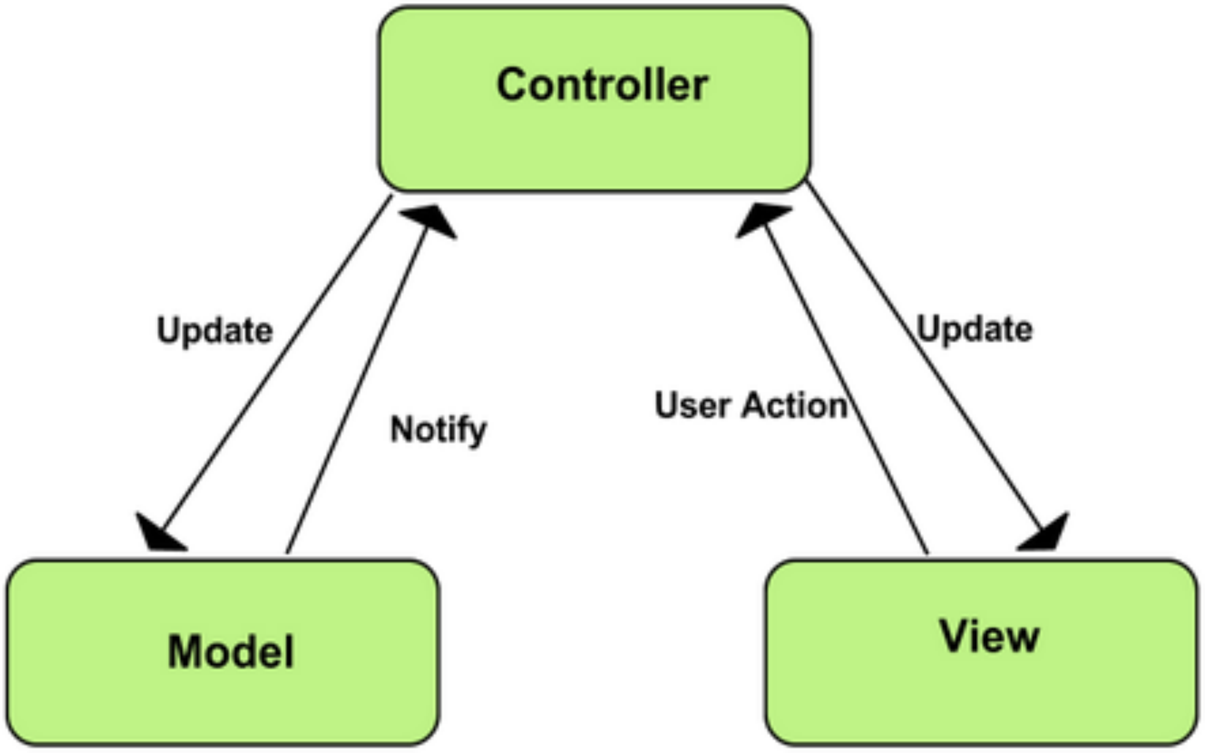
MV* Framework and Components

Goals

- ✓ Understanding MVC
- ✓ Understanding MV* structure of AngularJS
- ✓ Understanding Two-way dynamic binding
- ✓ Understanding Single-Page-Applications
- ✓ Understanding Declarative Approach of AngularJS
- ✓ Advantages of using AngularJS
- ✓ Component in AngularJS

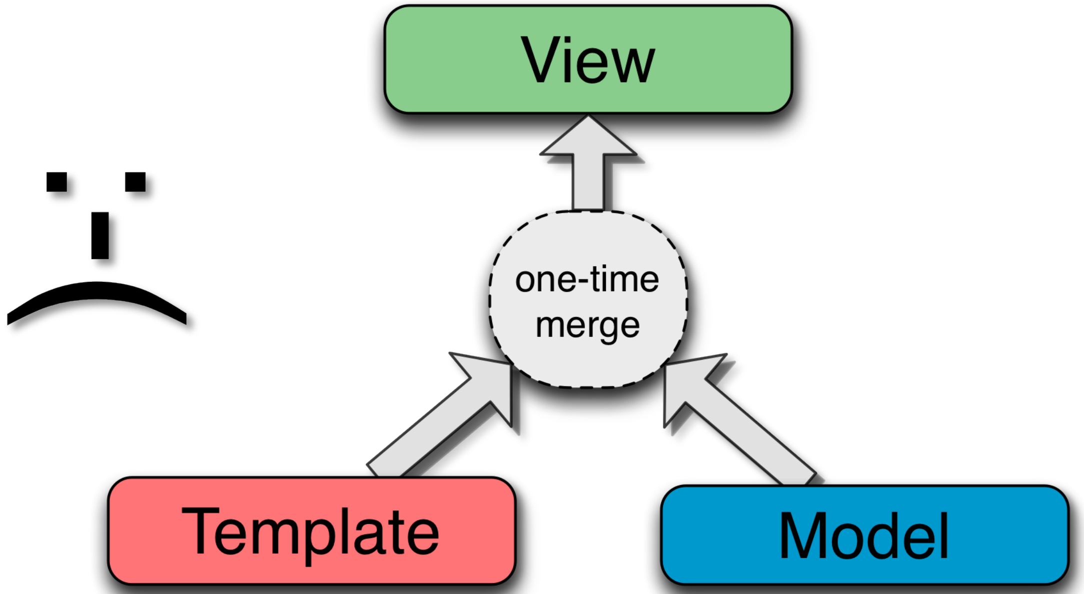
MVC frameworks

What is MVC



In a typical MVC framework, developer has to code the update mechanism for view or model. However **AngularJS** framework automatically updates model (data) or view according to changes made in any one of them.

Problem of MVC



MVC frameworks generally combines model with template and generate view. This approach has a major drawback when it comes to updating. This is **One-way Binding**

YOUSTART

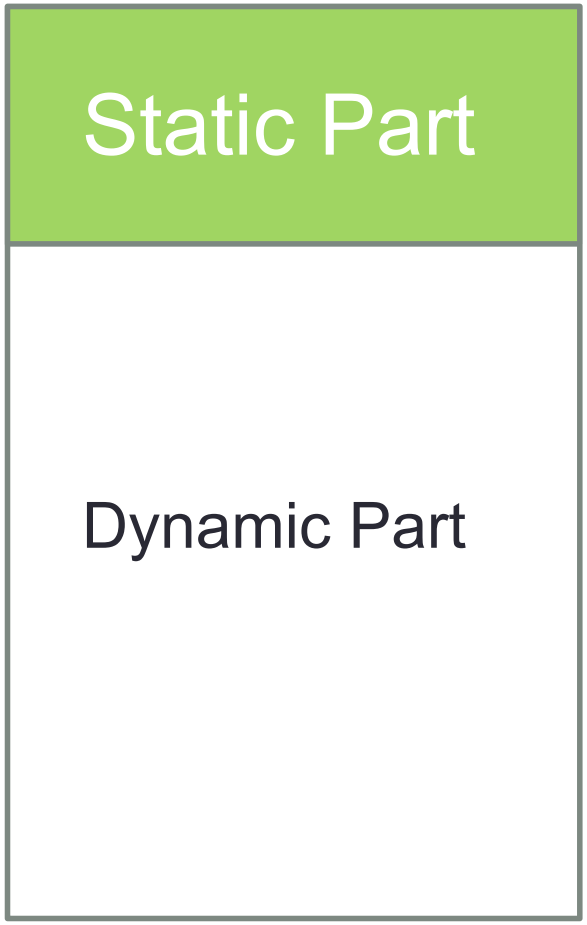
AngularJS

AngularJS

AngularJS is an open-source Javascript MVC (or better MV*) framework created by Google to build dynamic web applications with a proper architecture. It enhances the capabilities of HTML and lets you create Single Page dynamic web apps.

The other popular JavaScript web frameworks are Backbone.js, Ember.js etc.

Single Page App (SPA)

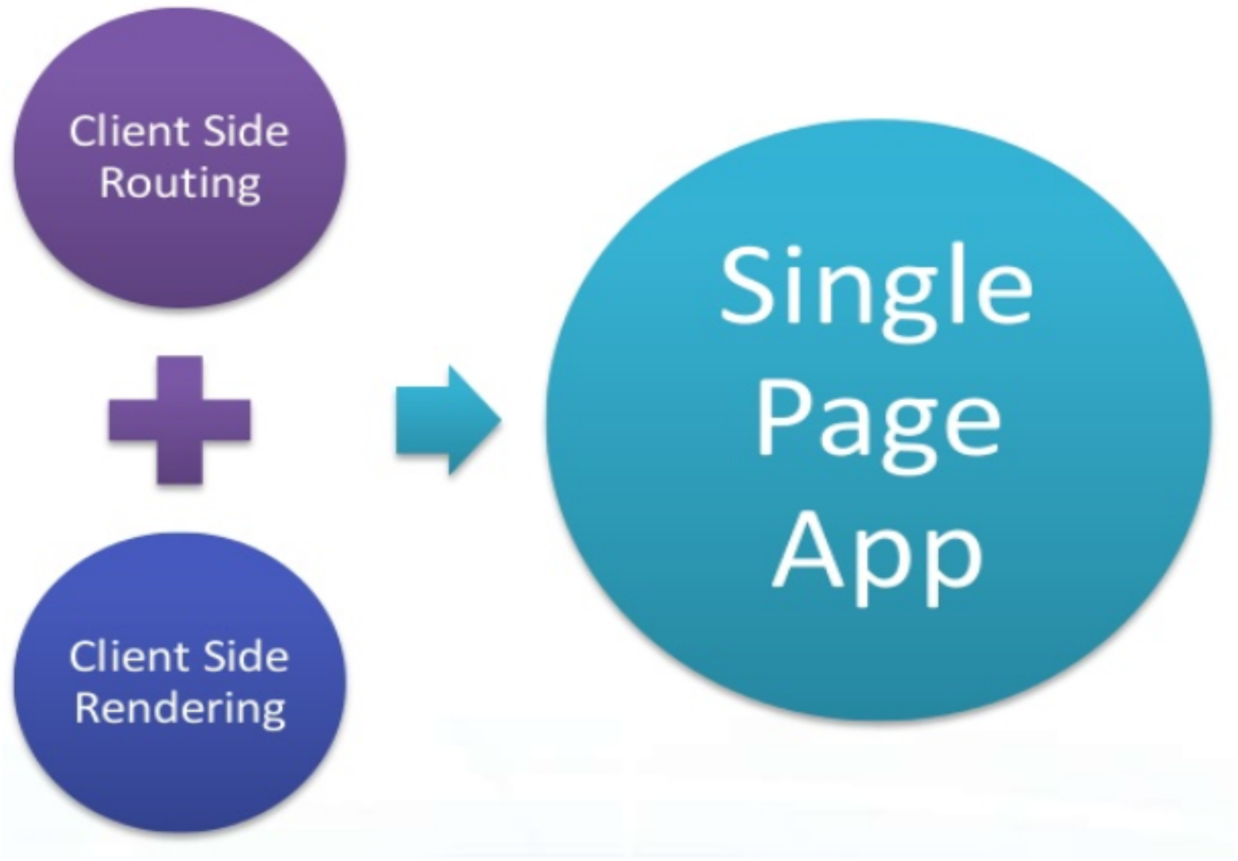


- Single page is loaded
- All script resources are loaded initially
- Partials are loaded depending on routing scheme
- Data remains persistent until reload

http://www.myapplication.com/login

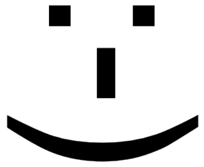
http://www.myapplication.com/profile

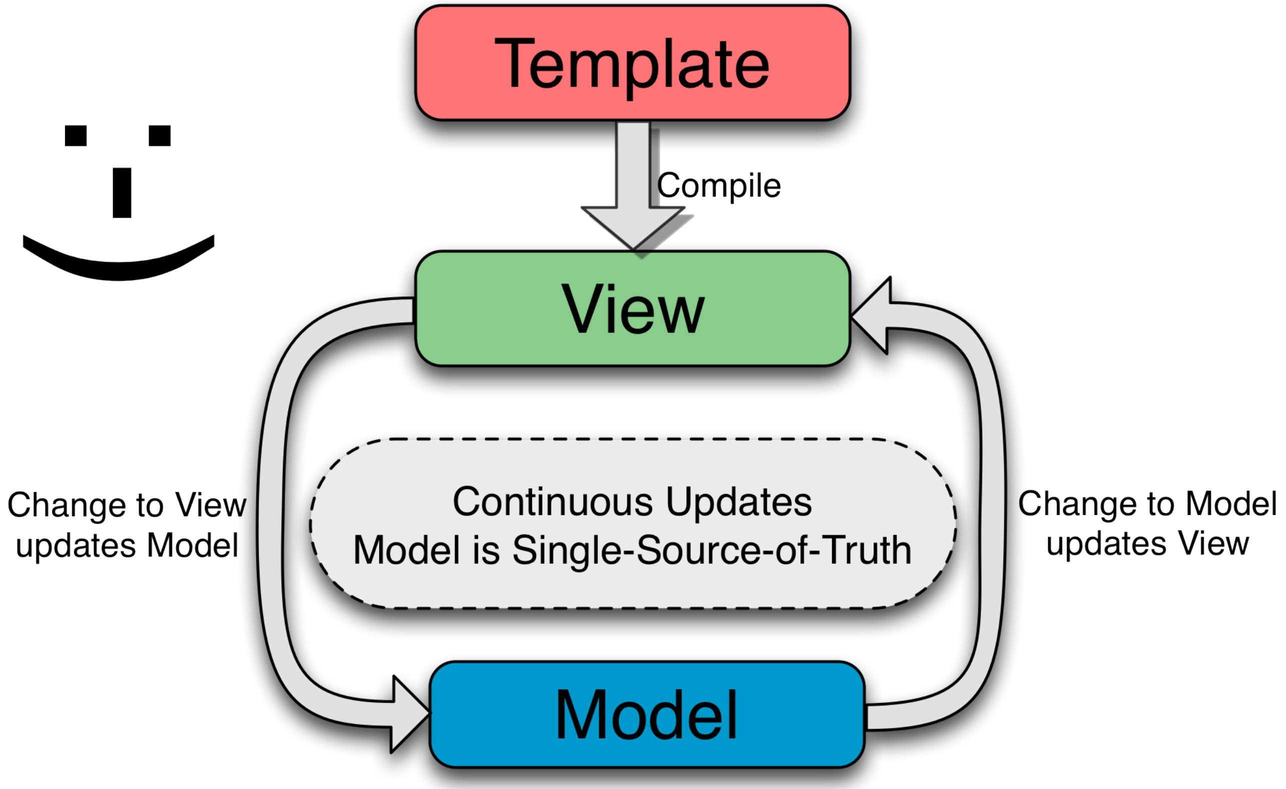
Single Page App (SPA)



A single-page application (SPA), is a web application or web site that fits on a single web page with the goal of providing a more fluid user experience akin to a desktop application.

AngularJS : 2-way binding

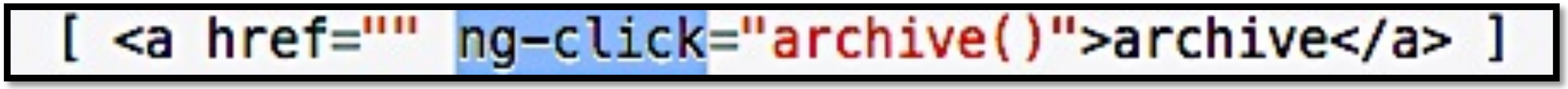



Template
Compile
View
Change to View updates Model
Continuous Updates
Model is Single-Source-of-Truth
Change to Model updates View
Model

AngularJS deals with the problem by continuously updating model and view on its own. This is **two-way data binding**

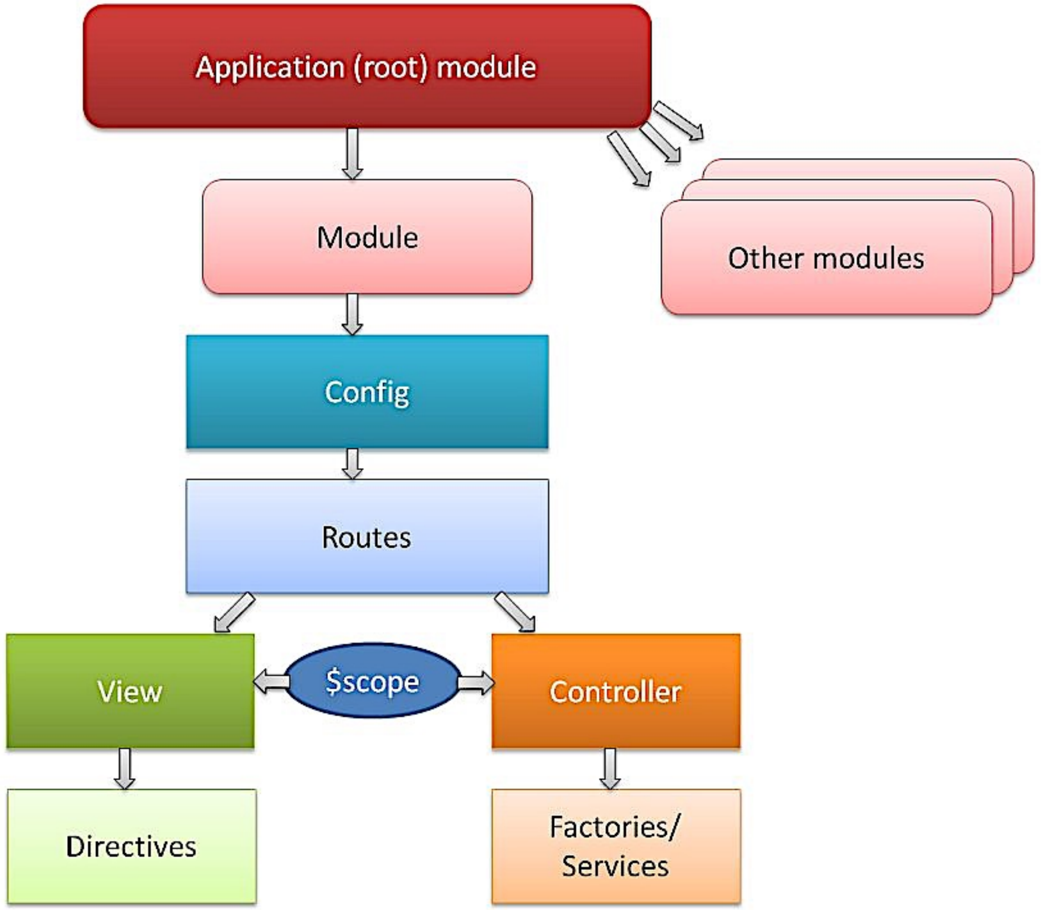
AngularJS : Declarative

DOM manipulation in AngularJS is controlled by directives which are identical to HTML tags and attributes in declaration. This **Declarative Approach** helps developer to understand the behavior of app easily.



```
[ <a href="" ng-click="archive()">archive</a> ]
```

AngularJS : Structure



YOUSTART

AngularJS App

App Initialization

App Initialization can we broken in multiple steps

1. **Embedding AngularJS library**
2. **Setting up ng-app**
3. **Binding ng-app with Angular Module**

Angular Core Library : angular.js
Angular Route Library : angular-route.js
Angular Cookies Library : angular-cookies.js

Full list: https://docs.angularjs.org/misc/downloading

Angular Module : ng-app

ng-app is an in-built directive which is used to

- **Initializing Angular code in HTML section**
- **Attaching a particular angular module**

```
<body ng-app  = “MyApp”>        - HTML

var app = angular.module ( “MyApp” , [] )        - JS
```

Data Model : ng-model

ng-model is an in-built directive which is used to

- **bind input data to scope of controller**

```
<input ng-model  = “data”>       - HTML

$scope.data             - JS
```

Angular Expressions

Angular Expression have different forms

- **When used in in-built directive**
- **When used anywhere in HTML**

```
<input ng-model  = “data”>

<p>{{data}}</p>

<img src={{data}} >
```

Controllers & Scope

Controllers : ng-controller

Angular Controller is container for logic in certain section

- **Multiple Controllers can be used**
- **App with no controllers also works**
- **Scope is bound to a particular controller**

```
<div ng-controller = “homeCtrl”>


</div>
<div ng-controller = “menuCtrl”>


</div>
```

Define Controllers

- Controllers are attached to angular module.
- Multiple controllers can be attached using chaining
- Controllers also maintain list of dependencies

```
.controller ( “homeCtrl” , function ($scope) {

$scope.data ;

})
.controller ( “menuCtrl” , function ($scope) {

$scope.data ;

})
```

$scope

$scope contains all properties and methods of a controller scope

```
.controller ( “homeCtrl” , function ($scope) {

$scope.data ;

})
.controller ( “menuCtrl” , function ($scope) {

$scope.data ;

})
```

YOUSTART

Thank You