INTRODUCING ANGULARJS

Ori calvo @2015

Objectives

- Understand the concept of Single Page Application
- How to write applications in AngularJS
- Get familiar with the basic building blocks of AngularJS

Topics

- □ What is a Single Page Application
- □ Other Frameworks
- □ Controller
- □ Scope
- □ Model
- □ Simple Directives

What is a Single Page Application

- A web application that fits on a single web page
- All HTML, JavaScript and CSS are retrieved with a single page load
  - Not a strict requirement
  - Can still load on demand
- The goal is to provide a more fluid user experience

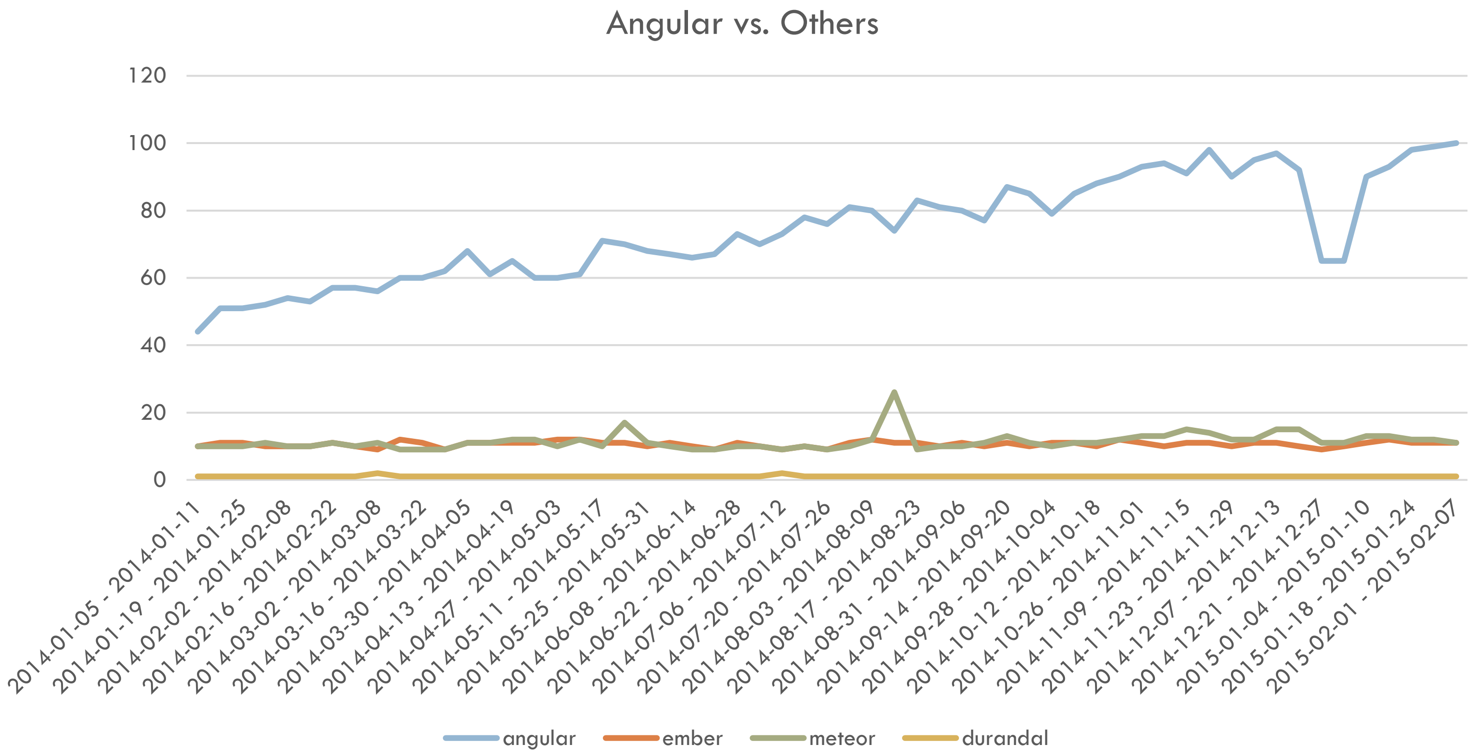
Single Page Application – Why

- No page reload
- No need to go to the server for HTML rendering
- Can cache local data easily
- SPA development resembles native application development
- MVC Frameworks
- Cross platform development
- Offline execution

Single Page Application - How

- Can manually integrate some building blocks
  - Object Oriented JavaScript
  - Client site template
  - Widget gallery
- Or, use a complete JavaScript framework
  - Angular
  - Durandal
  - Ember
  - Meteor

Client Side Frameworks


Angular vs. Others

What is AngularJS?

- Open source JavaScript framework
- Maintained by Google
- Assists with running SPAs
- MVC capabilities
- Supports IE9+
- Has managed to attract a lot of attention
- Good documentation

Main Characteristics

- Uses HTML as the templating language
- Doesn’t require an explicit DOM refresh
- Has an extensible components subsystem
- Supports dependency injection
- Supports declarative programming for building UI (directives)
- Promotes testing practices
- Has strong opinion about how to manage application logic vs. presentation
- Has no opinion about how to write classes and objects
- No widget gallery ☹

Angular 2

- Next version
- Not TRM yet
- A complete rewrite of Angular
- Not backward compatible
- Same concepts
- Component based
- Uses TypeScript, ES6 modules, Observables, ZoneJS and more …

Getting Started

- Download Angular script from http://angularjs.org/
  - You may want to download additional modules through the **extras** link
  - jQuery is not required
- Add a script reference to your HTML
- Add ng-app attribute to the HTML tag
- Start using Angular's directives like ng-model, ng-controller

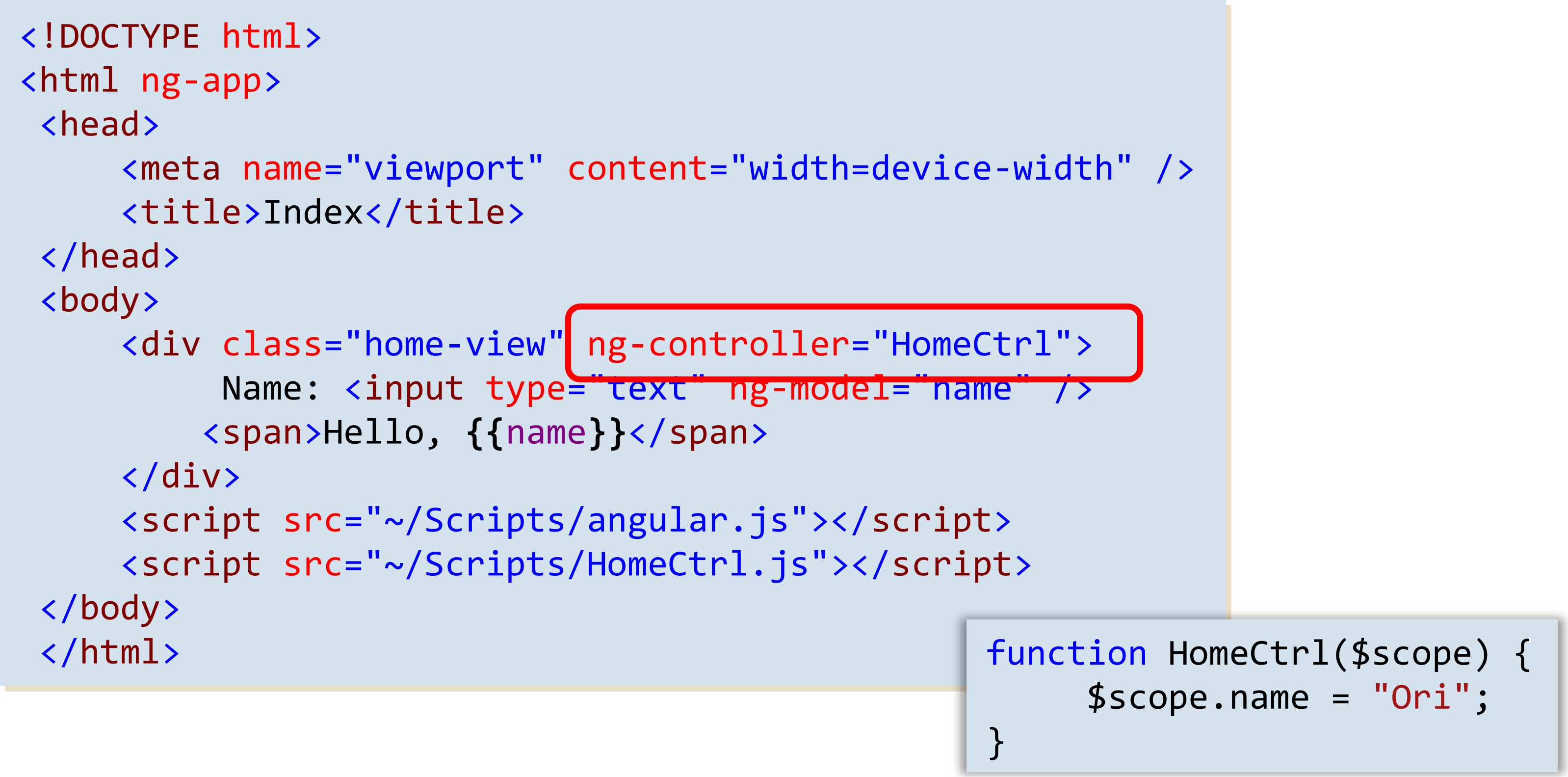
Hello World

```
<!DOCTYPE html>
 <html ng-app>
 <head>
     <meta name="viewport" content="width=device-width" />
     <title>Index</title>
 </head>
 <body>
     <div class="home-view">
          Name: <input type="text" ng-model="name" />
         <span>Hello, {{name}}</span>
     </div>
     <script src="~/Scripts/angular.js"></script>
 </body>
 </html>
```

Hello World – High Level Details

- The np-app directive is responsible for bootstrapping Angular
  - Remove it and the magic is lost
- The ng-model directive binds an input element to a field named name
  - This is a two way data binding
- {{name}} is an interpolation expression
  - Responsible for rendering the value of the name field
  - This is a one way data binding
- But what if we want to manipulate the field from code?
  - See next slide about Controller

Controller



```
<!DOCTYPE html>
<html ng-app>
 <head>
     <meta name="viewport" content="width=device-width" />
     <title>Index</title>
 </head>
 <body>
     <div class="home-view" ng-controller="HomeCtrl">
          Name: <input type="text" ng-model="name" />
         <span>Hello, {{name}}</span>
     </div>
     <script src="~/Scripts/angular.js"></script>
     <script src="~/Scripts/HomeCtrl.js"></script>
 </body>
 </html>
```

```
function HomeCtrl($scope) {
     $scope.name = "Ori";
}
```

Controller

- ng-controller directive binds a fragment of HTML with a JavaScript class known as the controller
- The controller is instantiated and receives a mysterious parameter named $scope
- The controller is responsible for manipulating the HTML and handle events
- Angular provides facilities to manipulate the HTML without actually accessing the DOM
  - $scope is the bridge
  - Controller becomes testable
  - Your responsibility to keep this model

Controller as Global Function

- □ Starting Angular 1.3 you can no longer define controller as a global function
- □ A module must be defined
- □ The controller must be associated with the module
- □ Will be covered later

$scope

- Is the glue between controller and view
- Exposes the domain model to the view
- Controller assigns properties to the $scope instance
  - The properties are available to the view
- Controller assigns methods to a $scope instance
  - The methods are available to the view
  - Usually are attached to DOM events
- Resembles a ViewModel from the MVVM pattern

Model

- The data attached to $scope is considered to be the model
- It is usually a simple JavaScript object(s)
  - Without any behavior
  - Just data
- Model does not require any framework's base class
  - This is unique to Angular !
- Model's property does not have to be primitive
  - Can hold a reference to other objects

Directive

- A marker on a DOM element
- When Angular bootstraps your application it traverses the DOM looking for directives
  - A process known as DOM compilation
- A directive attaches a specified behavior to a DOM element
- Angular comes with a set of built-in directives
- You can create your own directives
  - Good place to encapsulate DOM manipulation logic

Built-in Directives

Name	Behavior
ngChange	Bind to the change event
ngClass	Changes CSS class
ngClick	Bind to the click event
ngHref	Define URL with expression outside of href
ngIf	Removes an element from the DOM
ngRepeat	Duplicates an element for each model object
ngShow/ngHide	Shows/Hides and element
ngSwitch	Switch between different DOM element according to model state
More …	

ng-repeat

```
angular.module("myApp").controller("HomeCtrl", function HomeCtrl($scope) {
     $scope.contacts = [
         { id: 1, name: "Ori" },
         { id: 2, name: "Udi" },
         { id: 3, name: "Tommy" },
     ];
});
```

```
<div class="home-view" ng-controller="HomeCtrl">
     <ul>
         <li ng-repeat="contact in contacts">
             <span>{{contact.id}}</span>
             <span>{{contact.name}}</span>
         </li>
     </ul>
</div>
```

Summary

- □ Angular is great for implementing SPAs
- □ Offers modern architecture
- □ Less code
- □ Separation of concerns
- □ Two way data binding
- □ Plain JavaScript models
- □ Sophisticated compilation mechanism