

Introduction to AngularJS

Jogesh K. Muppala



THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系



香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

AngularJS History

- Designed by Misko Hevery
- First released in June 2012
- One of the most popular front-end JS frameworks

What is Angular?

- Structural framework for dynamic web applications:
 - HTML only does static documents
 - Angular fills in the gap to support dynamic applications
 - Solving the impedance mismatch
 - Designed with CRUD applications (data-driven) in mind
 - Declarative approach

Angular Vocabulary

- Two-way Data Binding
- Scope
- Directives
- Templates
- Routing
- Testing
- Modules
- Controllers
- Filters
- Factory
- Service
- Provider

Getting Started using Angular

- Include the Angular JS code

```
<script src="scripts/angular.min.js"></script>
```

Angular Built-in Directives

- HTML custom attributes:
 - data-* attributes (Bootstrap/jQuery)
 - Angular: ng-* attributes / data-ng-*
- Example: ng-app, ng-bind, ng-model, ng-init, ng-repeat etc.
 - They are also referred to using camelCase as ngApp, ngBind, ngModel, ngInit, ngRepeat etc.
 - Dash-delimited attributes are used since HTML is case insensitive
- Declarative programming in action
 - Declaratively call JavaScript functions

The ngApp Directive

- The ngApp (ng-app) directive is applied to specify the root of the application
 - Applying to <html> tag means entire page is under the control of the Angular application

The ngInit Directive

- The ngInit (ng-init) directive is used to
 - Evaluate an *expression*
 - Initialize a JavaScript variable

- Example:

```
<p ng-init="index = 1"> <p>  
  <div class="row row-content"  
    ng-init="dish={ name:'Uthapizza', ... } "> </div>
```


Angular Expressions

- Simple JavaScript Expressions
 - Evaluated against an Angular *scope* object
 - No conditionals, loops or exceptions
 - Expressions enclosed in `{{ expression }}`

- Example

```
<p> 6 + 5 = {{ 6 + 5 }} </p>
```

```
<div class="media-body">  
  <h2 class="media-heading"> {{dish.name}} </h2>  
  <p> {{dish.description}} </p>  
</div>
```

The ngModel Directive

- The ngModel directive binds the input value to a variable within the scope
 - Two-way data binding

- Example:

```
<div class="media-body">  
  ...  
  <p> {{dish.description}} </p>  
  <p>Comment: {{dish.comment}} </p>  
  <p>Type your comment:  
    <input type="text" ng-model="dish.comment"></p>  
</div>
```

Two-way Data Binding

- Binding an HTML or CSS property to a JavaScript variable
 - When the value of the variable is updated, the HTML/CSS property is also updated
 - Also works vice-versa with `<input>` fields

The ngRepeat Directive

- The ngRepeat directive is a looping construct
 - Loops over items in a collection
 - Instantiates a *template* for each item

- Example

```
<ul class="media-list"  
  <li class="media" ng-repeat="dish in dishes">  
    ...  
  </li>  
</ul>
```

Exercise: Introduction to AngularJS

- Use Angular directives like ng-app, ng-init, ng-model and ng-repeat
- Use Angular expressions in constructing a web template