Thursday, April 18, 2019

Angular

AngularJS

- Introduction
  - 它不是JavaScript库，里面没有我们可以直接调用和使用的function
  - 它不是像jQuery这样的DOM操作库， 但它使用jQuery的子集进行DOM操作（称为jqLite）
  - 更多地关注Web应用程序的HTML方面，适用于MVC / MVVM设计模式
  - AngularJS是由Google创建的Javascript MVC框架，用于构建适当的体系结构和可维护的Web应用程序。
- Philosophy
  - 如果HTML是为Web应用程序开发而设计的，ANGULARJS就是HTML
  - ANGULARJS围绕着这样的理念构建，当同时构建UI并将Web应用程序的不同组件连接在一起的时候，即声明性代码（declarative code）优于命令式代码（imperative code ）
- 例子

```
<!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>
```

- 为什么要用Angular
  - 定义了在客户端组织Web应用程序的多种方法
  - 通过在HTML中附加指令，自定义标记，属性，表达式，模板来增强HTML
  - 鼓励TDD
  - 鼓励MVC / MVVM设计模式
  - 代码重用
  - 适用于单页应用（SPA）
  - 酷功能
    - 声明性HTML方法，简单的数据绑定：双向数据绑定，可重用组件，MVC / MVVM设计模式，依赖注入，端到端集成测试/单元测试，路由，模板化，模块，服务，表达，过滤器，指令，表单验证，$ scope，$ http，$ routeProvider ……
- ng-app
  - 使用此指令自动引导应用程序
  - 每个HTML文档只能使用一个ng-app指令：<html ng-app>
- HTML compiler
  - Angular的HTML编译器允许开发人员教浏览器新的HTML语法。编译器允许您将行为附加到任何HTML元素或属性，甚至可以使用自定义行为创建新的HTML元素或属性。Angular调用这些行为扩展指令。
  - 编译器是一个angular服务，它遍历DOM寻找属性。 编译过程分两个阶段进行：

  - compile：遍历DOM并收集所有指令。 结果是linking function
  - link：将directives与scope组合起来并生成live view。scope中的任何更改都会反映在view中，并且与视图与用户的任何交互都会反映在scope中。这使scope成为唯一可以信任的来源

- Directive（指示）
  - 指令可以放在元素name，attribute，class name以及comments中
  - 指令是教HTML新技巧的一种方式。
  - 指令只是编译器在DOM中遇到它时执行的函数
  - 例子

```
<input ng-model='name'>
        Custom Defined Directives
<span draggable>Drag ME</span>
```

- Expression
  - 表达式是类似JavaScript的代码片段，通常放在bindings中，例如{{expression}}

    <BODY>1 + 2= {{1+2}}</ BODY>
- Forms
  - forms和controls提供validation services，以便可以通知用户无效输入。这提供了更好的用户体验，因为用户可以获得有关如何纠正错误的即时反馈。

```
<input type="text" ng-model="user.name" name="uName" required />
<button ng-click="update(user)“ ng-disabled="form.$invalid || isUnchanged(user)”>
 SAVE
</button>
```

- Module
  - 模块以声明方式指定应如何引导应用程序
  - 应用程序中可以有多个模块，这些模块也可以相互依赖。
  - 例子

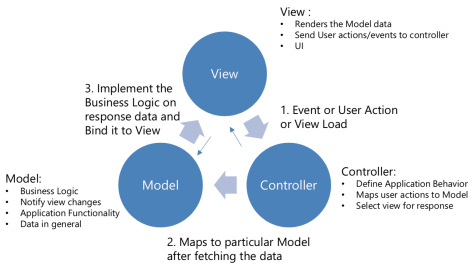
```
// declare a module
var myAppModule = angular.module('myApp', [--here goes the dependent Modules--]);
```

  - 模块配置有routes，controller，modules等。
- Routing
  - 它用于将URL深层链接到控制器和视图（HTML部分）。它监视$ location.url（）并尝试将路径映射到现有route definition

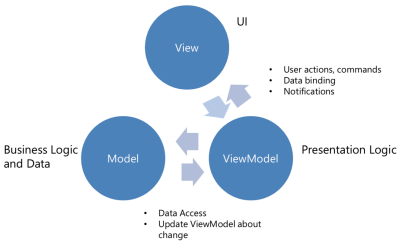
```
$routeProvider.when(‘/Book', { template: 'examples/book.html', controller: BookCntl, });
$routeProvider.when('/Book/chapter01', {
template: 'examples/chapter01.html',
controller: ChapterCntl, });
```

- Scope
  - Scope是一个引用应用程序model的对象。 它是表达式的执行上下文。
  - scope按层次结构排列，模仿应用程序的DOM结构。
  - scope可以观察表达式和传播事件。实际上是MVVM的ViewModel。
  - $Scope
- Filters
  - angular filters格式化数据以显示给用户
  - 格式：
    - {{expression [| filter_name [： parameter_value] ...]}}
    - {{uppercase_expression | uppercase }}
    - {{expression | filter1 | filter2}}
  - 可以创建自定义filter
- MVC & MVVM

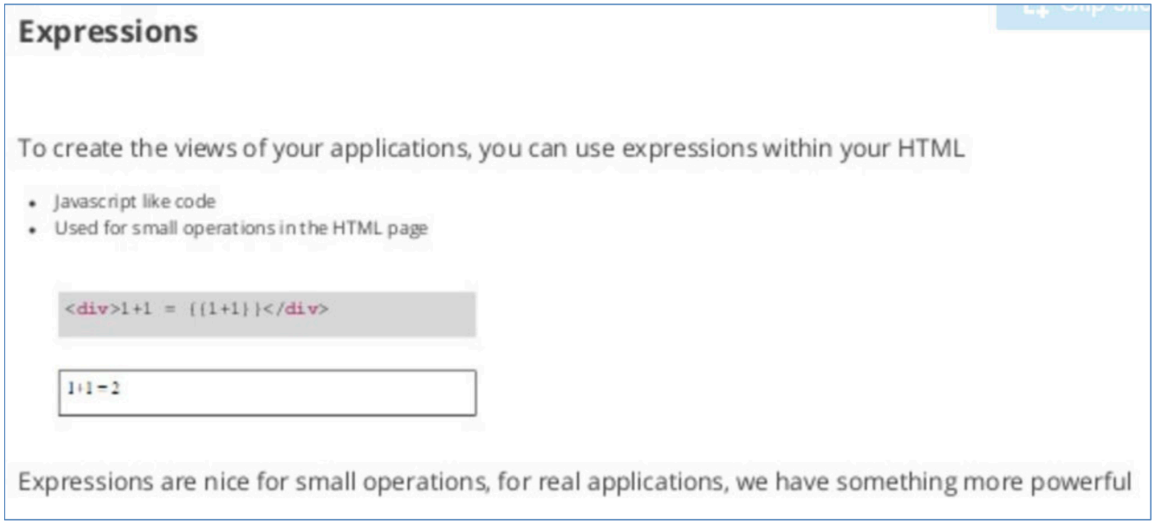
MVC : Model View Controller



MVVM: Model View ViewModel



例子



Directives

Extends HTML to structure your application

- Declarative
- Use the data available in the scope (more on that later)
- Create the DOM of the fly

Let's have a look at an example: ngRepeat.

```
<div>
    <div ng-repeat="user in users">
        <h3>{{user.name}}</h3>
        <p>{{user.description}}</p>
    </div>
</div>
```

It iterates on a collection in the scope to create the DOM

Directives - ngRepeat

```
<div>
    <div ng-repeat="user in users">
        <h3>{{user.name}}</h3>
        <p>{{user.description}}</p>
    </div>
</div>
```

For each elements in the collection "users" a new has been created with all its children

Directives - ngShow

AngularJS comes with a collection of standard directives that can be combined

```
<div>
    <div ng-repeat="user in users" ng-show="user.gender == 'female'">
        <h3>{{user.name}}</h3>
        <p>{{user.description}}</p>
    </div>
</div>
```

ngShow let you hide elements that do not validate a given predicate
Here, the users that do not have the gender "female" have generated a hidden

Directives - ngSwitch

AngularJS also provides you with complex directives like ngSwitch

```
<div>
    <div ng-repeat="user in users"
         ng-show="user.gender == 'female'"
         ng-switch="user.house">
        <h3>{{user.name}}</h3>
        <p>{{user.description}}</p>
        Sigil:
        <img src="images/targaryen.gif" ng-switch-when="Targaryen">
        <img src="images/stark.gif" ng-switch-when="Stark">
        <img src="images/lannister.gif" ng-switch-when="Lannister">
    </div>
</div>
```

With those directives, you can create the basic structure of your web application easily

Directives

A final word on directives

- All the directives of the AngularJS standard library are named "ngMyAwesomeDirective"
- You can use them with "ng-my-awesome-directive"
- Some directives can be used as attributes, comments, DOM elements name or even CSS classes

```
<div>
    <div ng-my-awesome-directive></div>
    <ng-my-awesome-directive></ng-my-awesome-directive>
    <div class="ng-my-awesome-directive"></div>
    <!-- directive: ng-my-awesome-directive -->
</div>
```

And of course, you can create your own directives (we will create a very basic one later)

Data Binding

connect your models and your views

Data Binding

Angular gives you the ability to define the binding between the data in your scope and your views

- Most directives that are using expressions are creating a bidirectionnal data binding for you
- You can create manually new bindings with the directive ngModel

```
<div>
    <div ng-repeat="user in users">
        <h3>{{user.name}}</h3>
        <p>{{user.description}}</p>
        Edit Description:
        <textarea rows="5" cols="50" ng-model="user.description">
    </div>
</div>
```

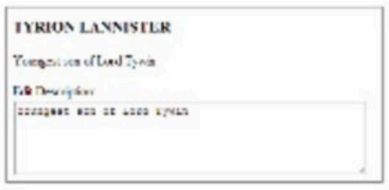
The changes are visible in real-time in all the expressions

Filters - uppercase

Angular comes with a collection of filters that can change the way your data are displayed

- Usage: {{expression | filter}}

```
<div>
    <div ng-repeat="user in users">
        <h3>{{user.name | uppercase}}</h3>
        <p>{{user.description}}</p>
        Edit Description:
        <textarea rows="5" cols="50" ng-model="user.description">
    </div>
</div>
```



You can also easily create your own filters

Filters

change the way your expressions are displayed