

概述：Angular与单页面应用程序

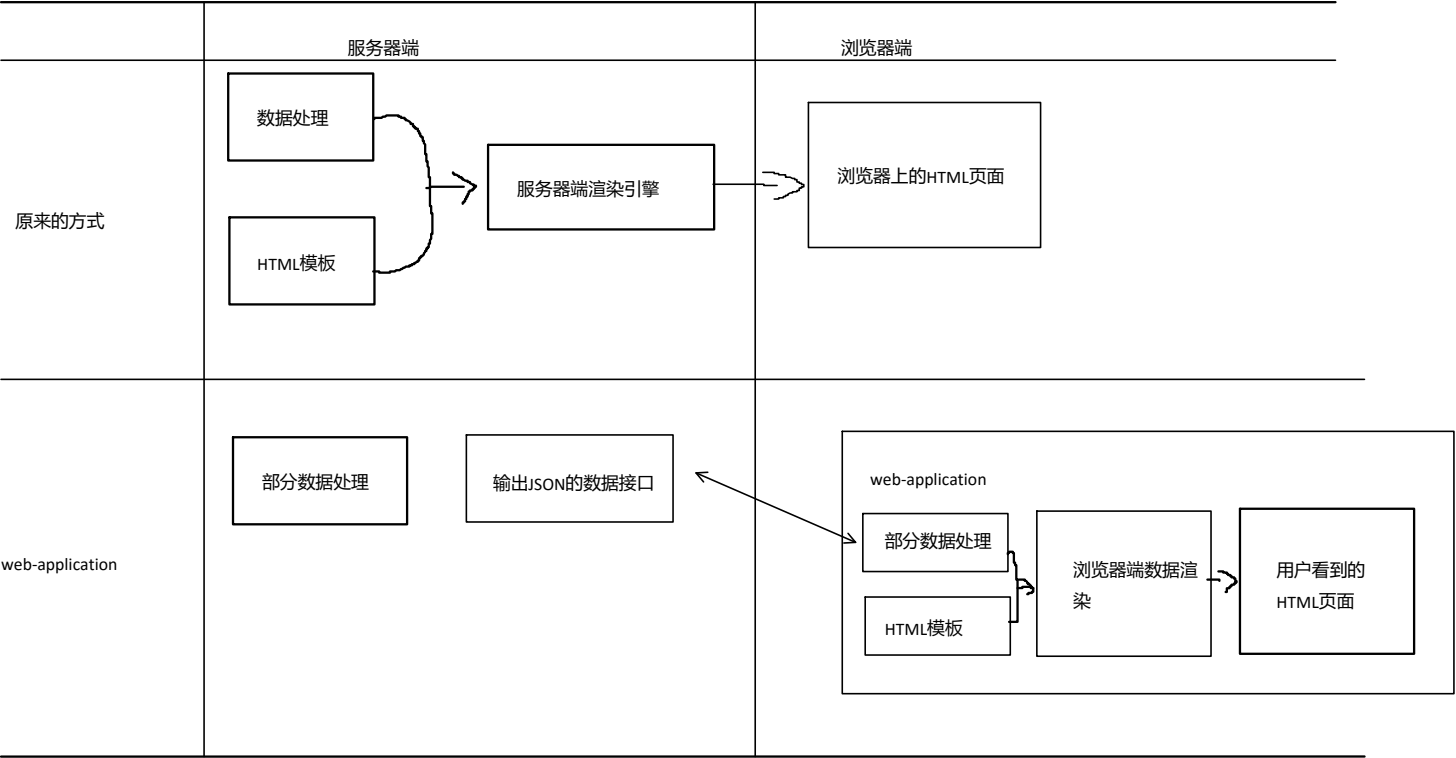


这是一段来自AngularJs官网的介绍。翻译如下：

HTML非常适合于描述静态的文档，但是如果我们试图用HTML构建web-application里面的动态的视图，它就有些无力了。AngularJS让你能够扩展HTML，构建更具表现力、可读性更高、开发更快捷的开发环境。

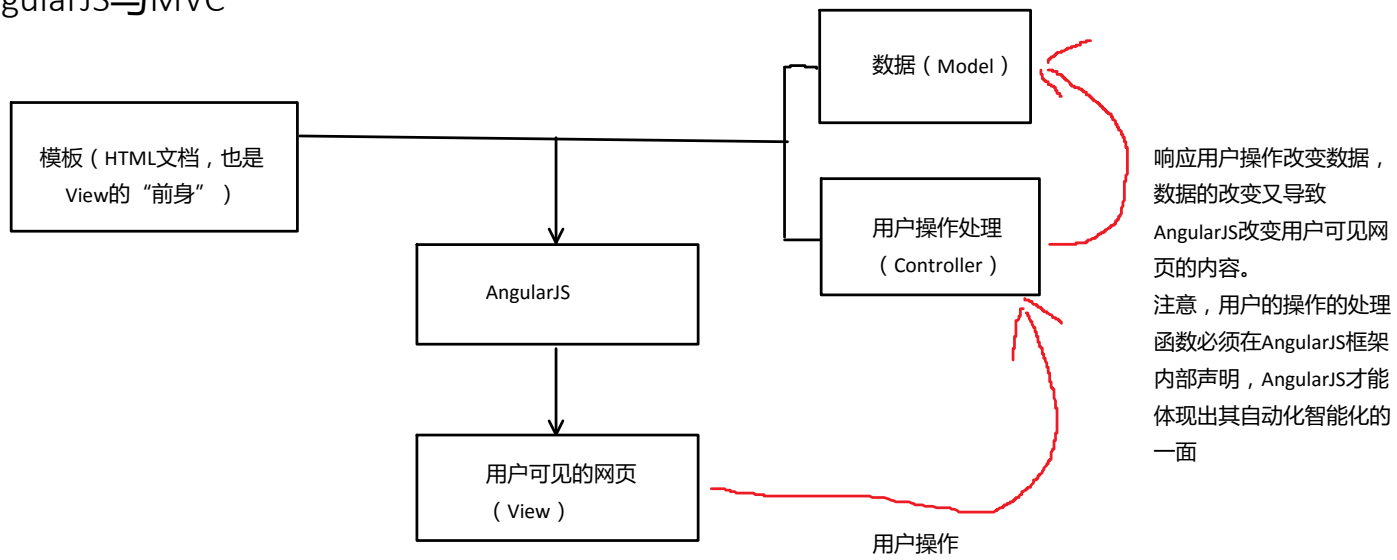
于是，什么是web-application？  
看下方表格。

web-application:





## AngularJS与MVC



# 常用指令：如何在文档上“打标记”

AngularJS的常用标记大致分成这样几类：

## 数据绑定类：

1. 插值语法：{{}}
2. 标签内容绑定：  
ng-bind
3. 表单控件双向绑定：  
ng-model

## 事件（动作）绑定类：

1. 点击事件：ng-click
2. 双击：ng-dblclick
3. 获得焦点：ng-focus
4. 失去焦点：ng-blur
5. 数据改变：ng-change

.....

更多可以查看文档。

## 加载处理：

1. ng-cloak
2. ng-src
3. ng-href

## 状态设置类：

1. ng-class
2. ng-style
3. ng-readonly
4. ng-disable
5. ng-hide,ng-show

## 程序结构类：

1. ng-app
2. ng-controller

## 仿流程控制类：

1. ng-if（如果）
2. ng-switch（分支判断）
3. ng-repeat（循环）（重点）

# AngularJS双向绑定

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>01AngularJS绑定示例</title>
  <!--引入AngularJS包-->
  <script src="../js/angular.js" ></script>
</head>
<!--用ng-app指定AngularJS运行于哪个标签上-->
<body ng-app>

  <input type="text" ng-model="demo">
  <div>{{ demo }}</div>

</body>
</html>
```

指定AngularJS框架运行在哪个标签的内部:ng-app属性

ng-model:

AngularJS内部有一个“JavaScript对象，用于存储数据”，ng-model指定这个对象上的一个属性，和这个标签，进行绑定：

用户有输入时，ng-model指定的那个属性，也随着用户输入进行更改。

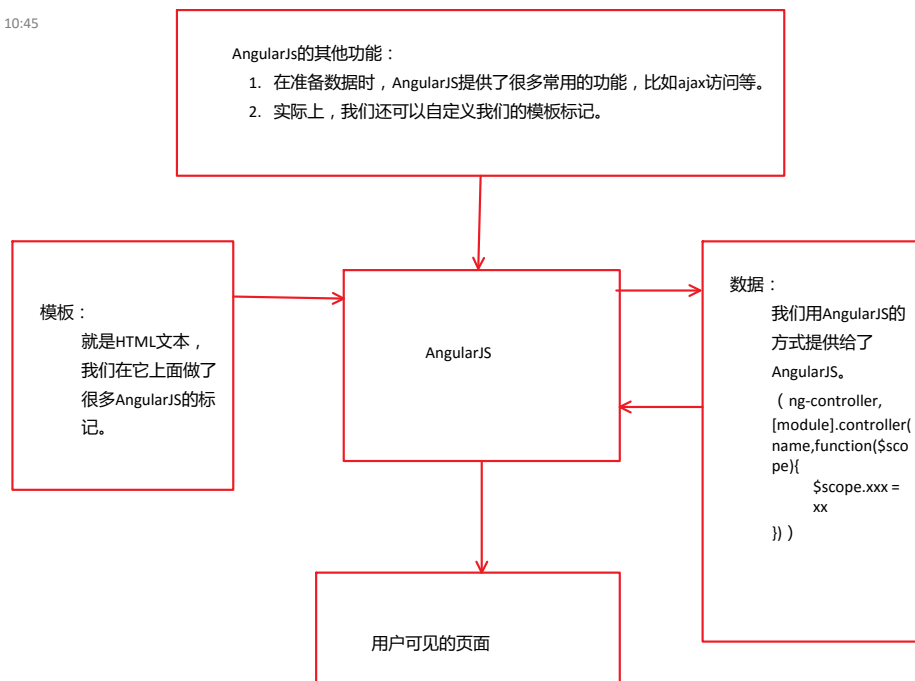
当那个属性内部的数据，因为其他原因有更改时，这个更改也会体现在这个标签上。

插值语法：

将model上的属性和这个插值语法进行绑定，网页上不显示插值语法，而是现实对应的数据

# AngularJS框架的大体结构

2016年5月11日 10:45



1. 如何使用AngularJS自带的标记。
2. AngularJS它的作用域、控制器，以及AngularJS的模块。
3. AngularJS提供的通用功能，主要要学的其实就是ajax。
4. 自定义AngularJS的通用功能。（服务，Service）AngularJS的依赖注入。
5. 如何自定义自己的模板标记。
6. AngularJS的“路由”。

# 今天上午的内容

2016年5月11日 10:37

1. 什么是单页面的应用：
  - a. 本质：无刷新的单页面，当用户有操作时ajax请求数据，局部的更新我们的页面
  - b. 理念：更好的用户体验，无停顿的用户交互
2. 单页面的应用的问题：
  - a. 写起来超级麻烦。
3. 所以有了AngularJS的框架：
  - a. 核心：双向数据绑定 -> 把数据和我们的HTML文档绑定起来，我们只需要关注数据就可以了。DOM操作由AngularJS去管理，一来方便，二来安全性高（程序员写代码出bug的概率变低）。

# 几个重要的概念：

2016年5月11日 11:49

1. 模板、数据的双向绑定。
2. 模块化：AngularJS指定了模块里面只能有某几种“部件”。（`[module].controller(...)`其实就是Controller部件）
3. 指令：directive，就是文档上打的各种“标记”。（新的标签名、或者新的属性名。）
4. 依赖注入：
  - a. jQuery提供功能时，怎么提供的？`$.ajax()` ... 这样，就是“把功能挂到\$对象上”
  - b. AngularJS提供功能时，怎么提供的呢？AngularJS中有大量的回调函数，在我们书写回调函数是，把参数名按正确的方式书写，AngularJS会自动地提供对应的“服务对象”。这个就是依赖注入。



# AngularJS Batarang

2016年5月11日 9:49

记得去谷歌的应用商店安装。用法是点enable，然后看scope内的内容。

# AngularJS模块和控制器的理解

AngularJS模块：把AngularJS的各种“零部件”组织到一起。

AngularJS的Controller：是AngularJS诸多零部件中的一种，功能是：提供\$scope，\$scope里有数据，也有方法（用于处理各种事件）。

我们一个AngularJS应用想要正常运行，它需要模板，也需要数据。

启动AngularJS程序时，需要指定一个模块，因为模块里面有各种数据之类的我们应用中用得着的东西。（ng-app）

AngularJS应用，需要数据，数据由Controller提供。Controller里面有一个作用域，Controller还有控制器的名字，然后用ng-controller指定这个Controller里面的\$scope在哪个元素内有效。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>05流程控制</title>
  <script src="../js/angular.js"></script>
</head>
```

<!--ng-app指定一个模块名-->

```
<body ng-app="demo">
```

我的程序要运行起来，必须有一个AngularJS的模块

<!--指定一个控制器在什么范围内生效（这里的控制器，指的是AngularJS的控制器）-->

```
<div ng-controller="mainController">
  {{name}}
</div>
```

这个\$scope对象，只在这段HTML（这个元素中）中有效。

```
<script>
  var app = angular.module('demo', []);
  app.controller('mainController', function ($scope) {
    $scope.name = 'main'
  })
</script>
```

AngularJS模块，它内部有一个控制器：mainController

```
</body>
</html>
```

# AngularJS模块与控制器

2016年5月9日 1:10

从今天起，不再使用ngHelper，而是使用AngularJS自己的API来构建AngularJS程序。首先看示例。

```
<!DOCTYPE html>
<html lang="en" ng-app="app">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body ng-controller="main">

  <h1>{{data.name}}</h1>
  <h2>年龄: {{data.age}}</h2>
  <h3>赞: {{data.praise}}</h3>

  <button ng-click="action.praise()">点赞</button>

<script src="../../node_modules/angular/angular.js"></script>

<script>
  var app = angular.module('app', []);

  app.controller('main', function($scope) {

    var data = $scope.data = {};
    data.name = '张三';
    data.age = 24;
    data.praise = 0;

    var action = $scope.action = {};
    action.praise = function () {
      data.praise += 1;
    }

  });
</script>
</body>
</html>
```

```
var app = angular.module('...', [...]);
```

是用来创建一个AngularJS模块的，它的返回值就是创建出来的AngularJS的模块本身。第一个参数是模块的名字，第二个参数是这个模块会依赖于的其他模块。（就相当于其他模块内部的控制

```
app.controller('...',function($scope){
  ....
})
```

是用来在模块上创建一个“AngularJS控制器”的，第一个参数是控制器名，第二个参数是控制器的工厂函数（用于创建并初始化这个控制器的函数），注意，这个函数接收一个参数名为\$scope

的参数，参数名不可以修改，这个\$scope用于和我们的模板相连接。和之前所说的MVC设计思想中的控制器不同，AngularJS框架因为兼容MVC、MVVM，所以控制器的功能和标准MVC并不相同。为了让它看上去更符合MVC的结构，我们认为的给\$scope加上了

AngularJS所有的零件都是属于模块的，所以AngularJS程序的书写通常是先创建模块，然后在模块上创建零件，然后多个模块互相组合，并最终组合到“启动模块”上——什么是启动模块呢？就是用ng-app指定的那个模块。ng-app的意思是，当前的AngularJS框架运行在哪段HTML代码中间，所依赖的启动模块是谁。如本例，就运行在整个HTML标签内部，依赖的启动模块是app。

而ng-controller的意思比较简单，就是指定一个控制器，与这个HTML标签相绑定，这样控制器的\$scope上的data和action就都可以应用在这个HTML标签的内部了。

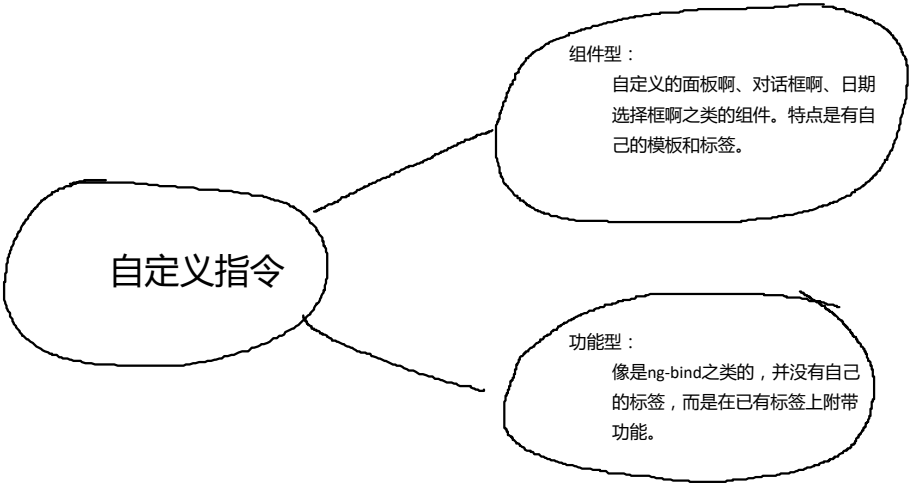
# AngularJS的指令

2016年5月9日 1:53

AngularJS的指令，其实就是指HTML模板上面的各种标记。第一天学的那些是AngularJS自带的标记，我们还可以自定义标记。

```
[...module].directive('icDemo',function(){
  return {
    .....
  }
})
```

- template/templateUrl：模板
- transclude：内嵌
- replace：替换
- scope：指令的作用域
- compile：编译期执行
- controller：指令内部的控制器，连接期执行，在link之前
- link：连接期执行



# AngularJS服务

2016年5月9日 2:03

在使用AngularJs的controller时，第二个参数是控制器工厂，可以用[...,'...',function(arg1,arg2,...){ ... }]的方式来声明这个工厂函数。其中的各个参数之前说是“AngularJS提供的各种功能”，实际上这个东西叫做AngularJS的服务。

我们也可以自己声明自定义的服务。

声明的方法很多：

1. [...module].factory('...', [...,'...', '...', '...',function(arg1,arg2,...){  
    return {  
        ....  
    }  
}])
2. [...module].service('...', [...,'...', '...', '...',function(arg1,arg2,...){  
    this.xxx = xxx;  
    ....  
}])
3. [...module].constant('...',value)
4. [...module].value('...',value)
5. [...module].provider('...', [...,'...', '...', '...',function(arg1,arg2,...){  
    return {  
        \$get : function(){  
            ....  
        },  
        ....  
    }  
}])

Angular程序的结构

