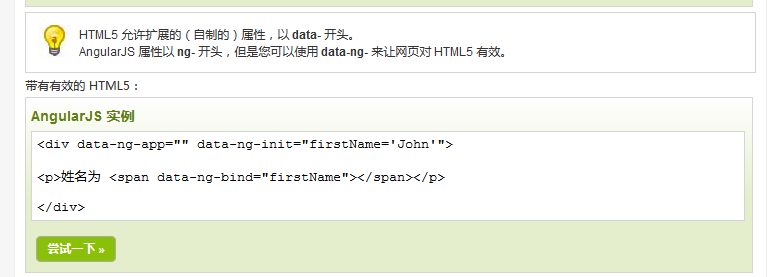
首先声明并不是说 所有的都要做博客，这篇是一个个人学习的笔记内容，并不能做博客文章。



|  |  |
| --- | --- |
|  | AngularJS 完美支持数据库的 CRUD（增加Create、读取Read、更新Update、删除Delete）应用程序。 把实例中的对象想象成数据库中的记录。 |

AngularJS 通过被称为指令的新属性来扩展HTML

AngularJS 通过内置的指令来为应用添加功能

AngularJS 允许你自定义指令

●ng-app指令告诉AngularJS，<div>元素是Angular应用程序的“所有者”

●一个网页可以包含多个运行在不同元素中的Angular应用程序

●{{firstName}}是通过ng-model=”firstName”进行同步的。

●ng-repeat指令会重复一个HTML元素

ng-app指令定义了AngularJS应用程序的根元素

ng-app指令在网页加载完毕时会自动引导(自动初始化)应用程序

ng-init指令为AngularJS应用程序定义了初始值

ng-model指定绑定HTML元素到应用程序数据

ng-model指令也可以：

1. 为应用程序数据提供类型验证(number、email、required)
2. 为应用程序数据提供状态(invalid、dirty、touched、error)
3. 为HTML元素提供CSS类
4. 绑定HTML元素到HTML表单

ng-repeat指令对于集合中(数组中)的每个项会克隆一次HTML元素

## 创建自定义的指令

除了 AngularJS 内置的指令外，我们还可以创建自定义指令。

你可以使用 **.directive** 函数来添加自定义的指令。

要调用自定义指令，HTMl 元素上需要添加自定义指令名。

使用驼峰法来命名一个指令， **runoobDirective**, 但在使用它时需要以 **-** 分割。

restrict值可以是以下几种：

E作为元素名使用

A作为属性使用

C作为类名使用

M作为注释使用

restrict默认值为EA,即可以通过元素名和属性名来调用指令

ng-model指令根据表单域的状态添加/移除以下类：

ng-empty

ng-not-empty

ng-touched

ng-untouched

ng-valid

ng-invalid

ng-dirty

ng-pending

ng-pristine

Scope(作用域)

Scope(作用域)是应用在HTML(视图)和JavaScript(控制器)之间的纽带。

Scope是一个对象,有可用的方法和属性

Scope可应用在视图和控制器上

Scope概述

AngularJS应用组成如下：

View(视图)、即HTML

Model(模型)、当前视图中可用的数据

Controller(控制器)、即JavaScript函数、可以添加或修改属性。

AngularJS应用程序被控制器控制

ng-controller指令定义了应用程序控制器

AngularJS过滤器可用于转换数据

currency

filter

lowercase

orderBy

uppercase

在AngularJS中，服务是一个函数或对象，可在你的AngularJS应用中使用。

在AngularJS内建了30多个服务。

有个$location服务,它可以返回当前页面的URL地址。

$http是AngularJS应用中最常用的服务。服务向服务器发送请求，应用响应服务器传送过来的数据。

AngularJS会一直监控应用，处理事件变化，AngularJS使用$location服务比使用window.location对象更好。

创建自定义服务

<div ng-app="myApp" ng-controller="myCtrl">

<p>255的16进制是</p>

<h3> {{hex}} </h3>

</div>

<script>

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

app.service('hexafy',function() {

this.myFunc = function(x) {

return x.toString(16);

}

});

app.controller('myCtrl',function($scope,hexafy){

$scope.hex = hexafy.myFunc(255);

});

</script>

ng-repeat指令是通过数组来循环HTML代码来创建下拉列表，但ng-options指令更适合创建下拉列表(哪怕数据源是对象也很方便操作)

ng-options=”x.site for x in sites”

ng-disabled指令绑定应用程序数据’mySwitch’到HTML的disabled属性。

ng-model指令绑定’mySwitch’到HTML input checkbox元素的内容(value)

ng-show指令根据value的值来显示(隐藏)HTML元素

你可以使用表达式来计算布尔值(true或false)。

AngularJS提供了很多内置的指令，你可以使用它们来为你的应用添加功能。

什么时候载入库？

对于与HTML应用程序，通常建议把多有的脚本都放置在<body>元素的最底部

这会提高网页加载速度，因为HTML加载不受制于脚本加载。

在我们的多个AngularJS实例中，你将看到AngularJS库是在文档的<head>区域被加载。

在我们的实例中，AngularJS在<head>元素中被加载，因为对angular.module的调用只能在库加载完成后才能进行。

另一个解决方案是在<body>元素中加载AngularJS库，但是必须放置在你的AngularJS脚本前面。

AngularJS表单是输入控件的集合。

以下HTML input元素被称为HTML控件：

input元素

select元素

button元素

textarea元素

<form novalidate>

...

</form>

novalidate属性在应用中不是必须的，但是你需要在AngularJS表单中使用，用于重写标准的HTML5验证。

AngularJS的首选样式表示Twitter Bootstrap，Twitter Bootstrap是目前最受欢迎的前端框架。

在AngularJS中，你可以在HTML中包含HTML文件。

大多数服务端脚本都支持包含文件功能(SSI:Server Side Includes)。

<div ng-include=”’myUsers\_List.html’”></div>

**数据绑定与模板**

数据到表现的绑定，主要是使用模板标记直接完成的：

{{ w }}

使用 {{ }} 这个标记，就可以直接引用，并绑定一个作用域内的变量。在实现上，ng 自动创建了一个 watcher。效果就是，不管因为什么，如果作用域的变量发生了改变，我们随时可以让响应的页面也随之改变。

模板到数据的绑定，只要是通过 ng-model 来完成的

**AngularJS提供了动画效果，可以配合CSS使用。**

AngularJS使用动画需要引入angular-animate.min.js库。

还需在应用中使用模型ngAnimate

<body ng-app=”ngAnimate”>

ngAnimate做了什么？

ngAnimate模型可以添加或移除class.

ngAniamte模型并不能使HTML元素产生动画，但是ngAnimate会监测事件，类似隐藏显示HTML元素，如果事件发生ngAnimate就会使用预定义的class来设置HTML元素的动画。

AngularJS依赖注入

什么是依赖注入？

维基上的解释是：依赖注入(Dependency Injection,简称DI)是一种软件设计模式，在这种模式下，一个或更多的依赖(或服务)被注入(或者通过引用传递)到一个独立的对象(或客户端)中，然后成为了该客户端状态的一部分。

该模式分离了客户端依赖本身行为的创建，这使得程序设计变得松耦合，并遵循了依赖反转和单一职责原则。与服务定位器模式形成直接对比的是，它允许客户端了解客户端如何使用该系统找到依赖。

一句话 --- 没事你不要来找我，有事我会去找你。

AngularJS提供了很好的依赖注入机制，一下5个核心组件用来作为依赖注入：

value -- 用于向控制器传递值

mainApp.value(‘defaultInput’,5);

factory --factory是一个函数用于返回值，在service和controller需要时创建。

mainApp.factory(‘MathService’,function(){

});

service

provider --AngularJS中通过provider创建一个service、factory等(配置阶段)。

Provider中提供了一个factory方法get()，它用于返回value/service/factory。

mainApp.config(function($provide){

$provide.provider(‘MathService’,function(){

this.$get = function() {

var factory = {};

factory.multiply = function(a,b){

return a\*b;

}

return factory;

};

});

});

constant --constant(常量)用来在配置阶段传递数值，注意这个常量在配置阶段是不可用的。

mainApp.constant(‘configParam’,’constant value’);

AngularJS路由运行我们通过不同的URL访问不同的内容。

通过AngularJS可以实现多视图的单页Web应用(single page web application,SPA)

通常我们的URL形式为http://yu.com/first/page,但在单页Web应用中AngularJS通过#+标记实现，例如：

<http://yu.com/#/first>

[http://yu.com/#/second](http://yu.com/#second)

当我们点击以上的任意一个链接时，向服务器端请求的地址都是一样的(http://yu.com/)。因为#号之后的内容在向服务端请求时会被浏览器忽略掉。所以我们就需要在客户端实现#号后面内容的功能实现。AngularJS路由就通过#+标记帮助我们区分不同的逻辑页面并将不同的页面绑定到对应的控制器上。