angular

标模指应初表迭绑 可显作控过服全选 事包过依路

通过<script>标签添加到html页面

模块定义了一个应用程序

var app = angular.module(“应用程序名”,[]);

在模块定义中[]参数用于定义模块的依赖关系。若模块没有依赖则为空，若有依赖则在中括号内写上依赖的模块名

js中应避免使用全局函数，因为容易被其他脚本文件覆盖。angularjs模块让所有函数的作用域在该模块下，避免了该问题。

指令是扩展的html属性，带有前缀ng-。可以使用data-ng-来对html5有效。

自定义指令

<a-b></a-b>

<script>

var app = angular.module(“应用程序名”,[]);

app.directive(”aB”,function(){

return{

restrict:”E”,

template:””

}

})

</script>

可以通过以下方式来调用指令，restrict默认为EA

E作为元素使用<a-b></a-b>

A作为属性使用<div a-b></div>

C作为类名使用<div class=”a-b”></div>

M作为注释使用<!-- directive:a-b -->

ng-app指令定义一个angularjs应用程序

ng-init指令初始化数据

ng-model指令用于表单，双向绑定

ng-repeat指令用于迭代，ng-repeat=”x in 数组名称”，显示序号$index

绑定表达式ng-bind=””指令或{{ }}

ng-disabled指令通过对错控制按钮是否可用

ng-show和ng-hide指令通过对错控制元素是否显示

$scope（作用域）是应用在html（视图）和js（控制器）之间的纽带，是一个对象，有可用的方法和属性。所有的应用都有一个$rootScope，它可以作用在ng-app指令包含的所有html元素中。是各个controller中scope的桥梁，可以在各个controller中使用

控制器

<div ng-app=”应用程序名” ng-controller=”控制器名”>

<input type=”text” ng-model=”firstName” /></br>

<input type=”text” ng-model=”lastName” /></br>

{{firstName+””+lastName}}

{{fullName()}}

</div>

<script>

var app=angular.module(”应用程序名”,[]);

app.controller(”控制器名”,function($scope,服务器名可有可无){

$scope.firstName=”John”;

$scope.lastName=”Doe”;

$scope.fullName=function(){

return $scope.firstName+””+$scope.lastName;

}

})

</script>

过滤器

自定义过滤器

app.filter(”过滤器名”,function(服务名可有可无){

return function(参数){

return

}

})

内建过滤器

<ul>

<li ng-repeat=”x in names|filter:engineer|orderBy:’country’”>

{{ (x.firstName|uppercase)+”,”+

(x.lastName|lowercase)+”,”+

(x.money|currency)

}}

</li>

</ul>

filter从数组中选择一个子集

orderBy根据某个表达式排列数组

lowercase格式化字符串为小写

uppercase格式化字符串为大写

currency格式化数字为货币格式

服务是一个函数或对象，可在应用中使用

自定义服务

app.service(”服务名”,function(){

this.方法名=function(){

}

})

内建服务

$location

$http

$http.get(”文件”).then(function(response){

})

$timeout

$timeout(function(){},毫秒)

$interval

$interval(function(){},毫秒)

全局api

angular.lowercase()转换字符串为小写

angular.uppercase()转换字符串为大写

angular.isString()判断对象是否为字符串，如果是返回true

angular.isNumber()判断对象是否为数字，如果是返回true

选择框，ng-options选择的值是一个对象，更好。ng-repeat选择的值是一个字符串

当数据源是数组

<select ng-model=”selectedName”>

<option ng-repeat=”x in names” value=”{{x.money}}”>

{{x.firstName}}

</option>

</select>

<select ng-model=”selectedName”

ng-options=”x.firstName for x in names” >

</select>

当数据源是对象

<select ng-model=”任意名代表选中值”

ng-options=”显示选项 for (键别名,值别名) in 对象” >

</select>

事件

ng-click=”方法名()”

$scope.方法名=function(){

}

包含

在html中，不支持包含html文件的功能

客户端包含

使用http请求（如ajax）从服务端获取数据，使用innerHTML写入到html元素中

angular包含

<div ng-include=”’html文件名’”></div>

跨域包含

默认情况下，ng-include指令不允许包含其他域名的文件。如果需要包含其他域名的文件，需要设置域名访问白名单

app.config(function($sceDelegateProvider){

$sceDelegateProvider.resourseUrlWhiteList([”允许的域名”]);

});

过渡和动画

angular使用过渡和动画需要引入angular-animate.min.js需在应用中使用模型ngAnimate

ng-app=”ngAnimate”或

var app=angular.module(”应用程序名”,[’ngAnimate’]);

过渡

html中有<div ng-hide=”变量名值为对或错”></div>

css中有

div{

background-color:red;

height:100px;

trasition:all linear 0.5s;

}

div.ng-hide{

height:0px;

}

动画

html中有<div ng-hide=”变量名值为对或错”></div>

css中有

div{

background-color:red;

height:100px;

}

@keyframes myChange{

from{

height:100px;

}to{

height:0px;

}

}

div.ng-hide{

animation:myChange 0.5s;

}

依赖注入（没事你不要来找我，有事我会去找你）5个核心组件value，provider，factory，constant，service

value用于配置阶段，向控制器传递值

var app=angular.module(”应用程序名”,[]);

app.value(”defaultInput”,1);

app.controller(”控制器名”,function($scope,defaultInput){

})

provider用于配置阶段，提供了$get()方法返回value/service/factory

var app=angular.module(”应用程序名”,[]);

app.config(function($provide){

$provide.provider(”myProvider”,function(){

this.$get=function(){

var factory={};

factory.multiply=function(a,b){

return a\*b;

}

return factory;

};

});

});

app.service(”myService”,function(myProvider){

})

factory是一个函数，用于返回值

var app=angular.module(”应用程序名”,[]);

app.factory(”myFactory”,function(){

var factory={}

factory.multiply=function(a,b){

return a\*b;

}

return factory;

})

app.service(”myService”,function(myFactory){

})

constant用于配置阶段，传递数值，这个常量在配置阶段不可用

app.constant(””,””);

路由

angularJS使用路由需要引入angular-route.js

[http://www.baidu.com/#/first](http://www.baidu.com/#first)

[http://www.baidu.com/#/second](http://www.baidu.com/#second)

点击以上的任意一个链接时，向服务端请的地址都是一样的（http://www.baidu.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。angularJS路由将不同的页面绑定到对应的控制器上

<ul>

<li><a href=”#/”>首页</a></li>

<li><a href=”#/first/”>first</a></li>

<li><a href=”#/second/”>second</a></li>

</ul>

<div ng-view></div>

var app=angular.module(”应用程序名”,[”ngRoute”]);

app.config(function($routeProvider){

$routeProvider

.when(”/”,{template:”首页页面”})

.when(”/first”,{template:”first页面”})

.when(”/second”,{template:”second页面”})

.otherwise({redirectTo:”/”});

})

路由配置对象

$routeProvider.when函数的第一个参数是url或者url正则规则，第二个参数为路由配置对象

$routeProvider.when(url,{

template:string类型，在ng-view中插入html内容，

templateURL:string类型，在ng-view中插入从服务端获取的模板文

件内容，

controller:function，string或数组类型，在当前模板上执行的

controller函数，生成新的scope，

controllerAs:string类型，为controller指定别名，

redirectTo:重定向的地址，

resolve:指定当前controller所依赖的其他模块

})