



第11讲 视图和路由

教学目标

1. 使用视图
2. URL路由
3. 配置路由

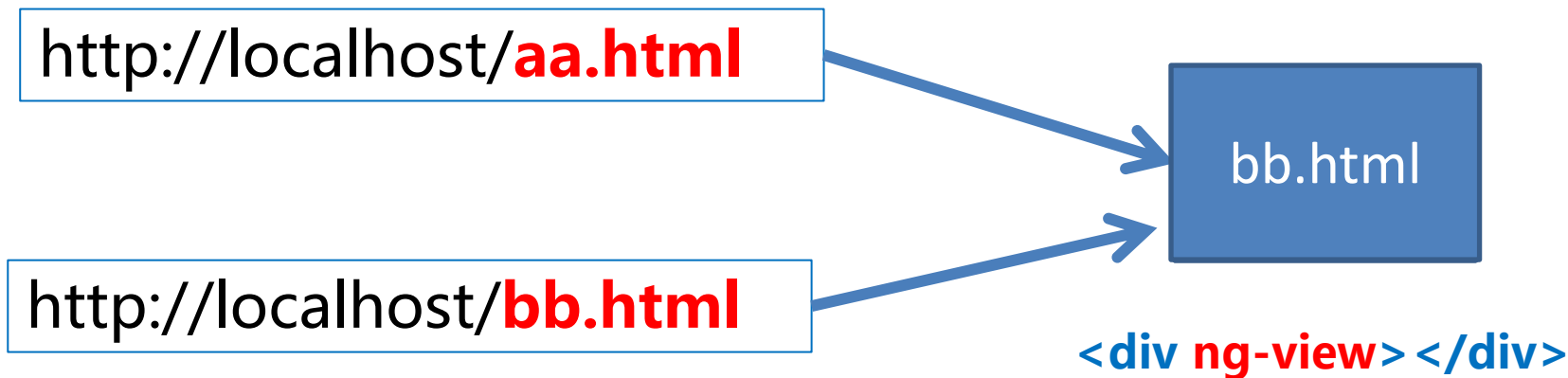
1

使用视图

①

实现原理

AngularJS提供 **ng-view**指令可以用来实现视图



ng-view的实现原理是根据路由的切换动态编译html模板。

序

多视图的单页应用

AngularJS提供 **ng-view** 和 **ng-template** 指令，以及 **\$routeProvider** 服务。支持通过在单个页面上的多个视图的单页应用。

②

ng-view

ng-view 指令只是简单地创建一个占位符，是一个相应的视图(**HTML或ng-template视图**)，可以根据配置来放置。

使用:

定义一个div与ng-view。

```
<div ng-app="demoApp">  
  <div ng-view> </div>  
</div>
```

Demo:listing 01.html

3

ng-template

ng-template 指令是用来创建使用script标签的HTML视图。它包含一个用于由\$routeProvider映射控制器视图“id”属性。

使用

定义类型作为主模块中 ng-template 的脚本块。

```
<div ng-app="mainApp">
  <script type="text/ng-template" id="add">
    <h2> Add Student </h2> {{message}}
  </script>
</div>
```

Demo:listing 01.html

4

\$routeProvider

\$routeProvider是一组**网址的配置**，将它们映射相应的HTML页面或 ng-template，并附加一个控制器使用相同的**服务**。

```
var mainApp = angular.module("mainApp", ['ngRoute']);  
  
mainApp.config(['$routeProvider',  
  
                function($routeProvider) { ..... }]);
```

Demo:listing 01.html

2

URL路由

1

AngularJS 路由

AngularJS 可以实现多视图的单页Web应用。

那么多个单页面之间如何访问呢？

那么同一页面多个不同内容之间如何访问呢？

AngularJS 路由允许通过不同的 URL 访问不同的内容。

1

AngularJS 路由

AngularJS的路由其实是一种纯前端的解决方案。不同于后端路由，它的本质其实是：**当请求一个url时，根据路由配置匹配这个url，然后请求模板片段，并插入到 ng-view 中去。**所以从某种意义上来说，AngularJS的路由通过改变url来进行页面的局部刷新。

2

ngRoute模块

AngularJS的**ngRoute** (**路由**)是一个单独的模块

名称	所属	作用
ngView	指令	定义路由模板的视图层
\$routeProvider	提供器	提供路由配置
\$route	服务	用于构建各个路由的url、view、controller这三者的关系
\$routeParams	服务	解析返回路由中带有的参数

②

安装ngRoute模块

AngularJS路由机制是靠ngRoute模块提供的，通过hash和history两种方式实现了路由，可以检测浏览器是否支持history来灵活调用相应的方式。

ngRoute模块既[angular-route.js](#)文件，必须将它下载到本地。

调用： [angular.js](#)文件之后调用

```
<script src="angular-route.js"></script>
```

Demo:listing 03.html

③

定义URL路由

\$route服务提供的核心功能是设置URL与视图文件名称之间的映射，被称为**URL路由**或就叫做**路由**。使用**\$route**服务的提供器**\$routeProvider**定义映射。



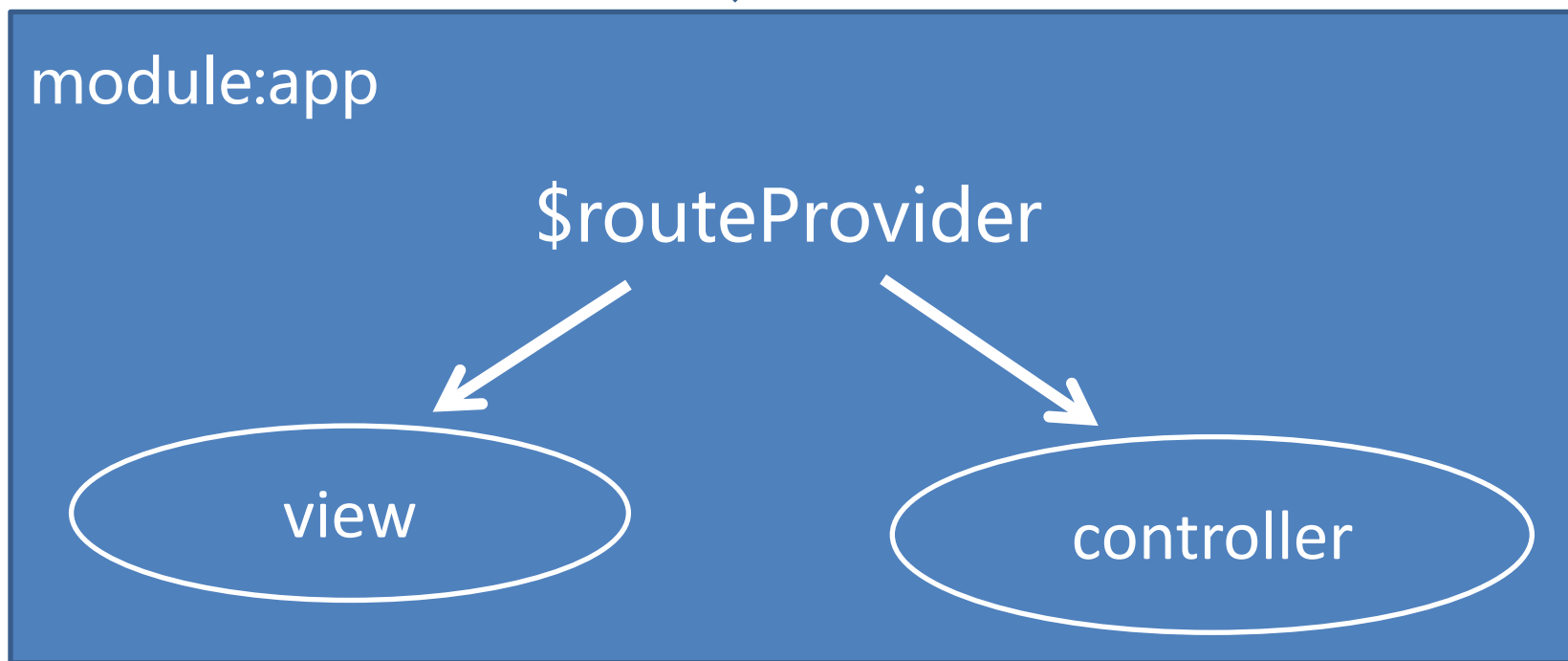
\$route服务的提供器

3

定义URL路由



请求.....



③

定义URL路由

通常的URL形式为 **http://***.com/first/page**，但在单页Web应用中 AngularJS 通过 **url + # + 标记** 实现，例如：

http:// localhost/#/

http:// localhost /#/aa

http:// localhost /#/bb

③

定义URL路由

必须在模块声明中注入对**ngRoute**的依赖

```
angular.module( 'DemoApp', [ 'ngRoute' ] )  
.config(function( $routeProvider ){ });
```

Demo:listing02.html

```
angular.module( 'DemoApp', [ 'ngRoute' ] )  
.config([ '$routeProvider', function( $routeProvider ){  
}]);
```

`$routeProvider`提供了定义路由表的服务，它有两个核心方法：

1、`when(path,route)`

path参数是string类型，表示该条路由规则所匹配的路径
route参数是object，用来指定当path匹配后所需的一系列配置项。

2、`otherwise(params)`

对应路径匹配不到时的情况，这时候我们可以配置一个**redirectTo**参数，让它重定向到404页面或者是首页。

route参数是object，用来指定当path匹配后所需的一系列配置项。

对象常用属性：

template: '视图模板内容'，

templateUrl: '外部视图模板路径'，

controller: '对应的控制器'，

Demo:listing03.html

4

定义路由表

```
angular.module( 'routingDemoApp' ,[ 'ngRoute' ])  
.config([ '$routeProvider' ,  
function($routeProvider){  
    $routeProvider  
        .when( '/' ,{template: '这是首页页面' })  
        .when( '/computers' ,{template: '这是公司简页面' })  
        .when( '/printers' ,{template: '这是新闻中心页面'})  
        .otherwise({redirectTo: '/'});  
}]);
```

Demo:listing04.html

每个 URL 都有对应的视图和控制器

```
<body ng-app="DemoApp" >  
<div ng-view></div>  
</body>
```

该 **div** 内的 **HTML** 内容会根据路由的变化而变化。

Demo:listing04.html

3

配置路由

①

使用路由参数

\$routeProvider . when(path , route)

route参数是一个object，常用的属性配置项：

- ◆ **template**
- ◆ **templateUrl**
- ◆ **Controller**
- ◆ **redirectTo**
- ◆ **resolve**

①

使用路由参数

route路由配置项

名称	描述
controller	指定与路由显示的视图关联的控制器的名称
controllerAs	指定控制器别名
template	指定视图内容。
templateUrl	指定路由所匹配显示的视图文件的URL
resolve	指定一组控制器的依赖
redirectTo	指定当路由匹配时浏览应重定向到的路径

2

route参数

◆ **template**

string或function类型，视图所用的模板，这部分内容将被
ngView引用

◆ **templateUrl**

string或function类型，当视图模板为单独的html文件
或是使用了<script type="text/ng-template">定义模板时使用

2

route参数——templateUrl

使用**templateUrl**加载外部模板文件

```
.config(['$routeProvider', function($provider1){  
    $provider1  
    .when('/', {template: '这是ex02.html页面'})  
    .when('/a', {templateUrl: 'aa.html'})  
    .when('/b', {templateUrl: 'bb.html'})  
    .otherwise({redirectTo: '/'});  
}]);
```

Demo:listing05.html

2

route参数

◆ controller

function或string类型。在当前模板上执行的controller函数，生成新的scope。

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

Demo:listing06.html

2

route参数——controller

```
angular.module('DemoApp', ['ngRoute'])
  .config(function($routeProvider){
    $routeProvider
      .when('/', {template: '这是当前页面'})
      .when('/a', {
        templateUrl: 'aa.html',
        controller: "aaa"
      })
      .when('/b', {
        templateUrl: 'bb.html',
        controller: "bbb"
      })
      .otherwise({redirectTo: '/'});
  })
```

Demo:listing06.html

2

route参数

route参数是一个object :

◆ redirectTo

重定向的地址

```
.otherwise({redirectTo: '/'});
```

◆ resolve

指定当前controller所依赖的其他模块

2

route参数

```
.config(function$routeProvider){
  $routeProvider
    .when('/', {template: '这是当前页面'})
    .when('/a', {templateUrl: 'aa.html'})
    .when('/b', {
      templateUrl: 'bb.html',
      resolve: {.....}
      //resolve完成状态信号返回后再执行路由
    })
    .otherwise({redirectTo: '/'});
})
```

Demo:listing07.html

3

访问路由和路由参数

在路由对应的控制器中，可以通过 **\$routeParams** 参数来获取这个url参数。

```
.controller('postId', function ($scope, $routeParams) {  
    $scope.msg = 'id: ' + $routeParams.postId;  
})
```

```
.when('/post:postId', {  
    templateUrl: 'bb.html',  
    controller: "postId"  
})
```

Demo:listing08.html

3

访问路由和路由参数

```
<script src="angular.js"></script>
<script src="angular-route.js"></script>
<script src="listing10.js"></script>

angular.module('DemoApp', ['route1', 'ngRoute'])
.config(function($routeProvider){
    $routeProvider
        .otherwise({redirectTo: '/'});
})
```

Demo:listing09.html

新闻发布系统

新闻列表

发布新闻

修改新闻

id	标题	内容	发布时间
1	title 1111	content 1111111	2017年03月29日 16:05:08
2	title 2222	content 2222222	2017年03月29日 15:22:41
3	title 3333	content 3333333	2017年03月29日 16:05:08
4	title 4444	content 4444444	2017年03月29日 16:05:08
3	title 5555	content 5555555	2017年03月29日 16:05:08
3	title 6666	content 6666666	2017年03月29日 16:05:08
3	title 7777	content 7777777	2017年03月29日 16:05:08

Demo:ex03.html

本课小结

1. 使用视图
2. URL路由
3. 配置路由

TNAKS

主讲：王智娟

QQ: 24132228

Email: wangzhijuan@onest.net