



# AngularJS 高级(三)

讲师：文元



泛IT职业教育机构

## 信·机构

- 慧科教育集团旗下泛IT职业教育培训机构。
- 与找座儿形成国内第一家O2O混合式教学落地基地。
- 良心教学，可信赖大机构！

## 新·模式

- 同时采用最合理的TS教学模式和最合理教学时间安排。
- 全程手把手教学、边学边练，7\*12小时答疑指导，保证学员即学即会。

## 心·服务

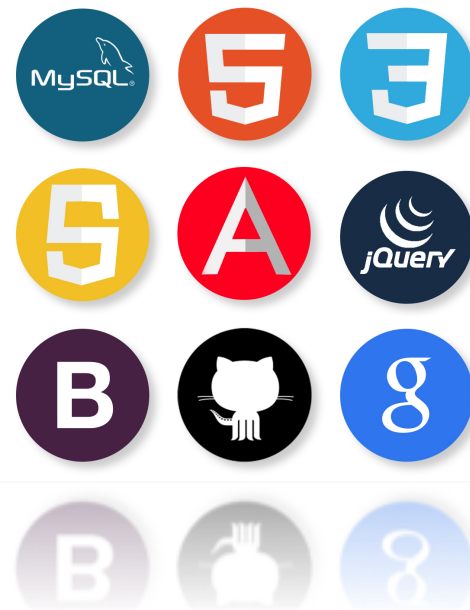
- 多年教学经验资深讲师倾心教学
- 配备专门的生活导师，提供入学接送、精选宿舍、生活指导、学习督促等一条龙服务。
- 提供专业就业指导服务，简历指导+模拟面试，保证快速就业。

## 薪·未来

- 无限互联学生毕业平均薪资12755元，比同行业机构平均高出12%。
- 98%的无限互联学生在毕业4周内成功就业！

# 本章知识点

- \$location
- 其他服务
- 路由



# 一、\$location

1. \$location 的介绍
2. \$location 的方法
3. \$location 对象事件

- 在开发 web 页面过程中,经常使用 location 用于刷新页面或跳转到某个指定的 URL 地址,在 Angular 中,这些功能都被优雅地封装到 \$location 服务中,借助这个服务提供的 API ,不仅可以实现在 JS 代码中 location 对象的所有功能,而且这个服务还与 Angular 框架集合在一起,使用方便
- 使用 \$location 服务不仅可以解析地址栏中的 URL 地址,还能将解析后的 URL 地址运用到 \$location 服务中,使地址栏中的 URL 和 \$location 服务在各自变化时产生互动效果
- \$location 服务中还内置了许多有用的方法和事件,还可以注入其他的服务,实现更加强大的功能

- 只读方法指的是只能执行方法,不能通过方法修改对象中的属性值
- `absUrl` : 该方法用于返回 URL 地址中编码后的完整内容,包含了 hash 片段、端口号和请求协议名称等
- `protocol` : 该方法用于返回 URL 地址中请求的协议名称,包括常见的 `http` 和 `https` 等
- `host` : 该方法用于返回 URL 地址中请求的主机名称,例如,URL 地址为 <http://www.wuxianedu.com>.,那么返回的内容是 [www.wuxianedu.com](http://www.wuxianedu.com)
- `port` : 该方法用于返回 URL 地址中请求的端口号,在通常情况下,默认的端口号为 80,如果 URL 地址为 `http://www.rttop.cn:9090/`,则执行方法后,返回内容是 9090

- 只读方法指的是只能执行方法,不能通过方法修改对象中的属性值
- `absUrl` : 该方法用于返回 URL 地址中编码后的完整内容,包含了 hash 片段、端口号和请求协议名称等
- `protocol` : 该方法用于返回 URL 地址中请求的协议名称,包括常见的 `http` 和 `https` 等
- `host` : 该方法用于返回 URL 地址中请求的主机名称,例如,URL 地址为 <http://www.wuxianedu.com>,那么返回的内容是 [www.wuxianedu.com](http://www.wuxianedu.com)
- `port` : 该方法用于返回 URL 地址中请求的端口号,在通常情况下,默认的端口号为 80,如果 URL 地址为 `http://www.rttop.cn:9090/`,则执行方法后,返回内容是 9090

- 读写方法指的是不仅可以通过执行方法获取对象中的属性,还可以通过方法传递的参数修改默认属性的值
- url : 不带参数时,返回 URL 地址栏中 # 后的内容,如果带参数,则修改地址栏中 # 后的内容,修改后返回一个 \$location 对象
- hash : 不带参数时,返回 URL 地址栏中 hash 片段内容,如果带参数,则修改地址栏中的 hash 片段内容,修改后返回一个 \$location 对象
- search : 不带参数时,返回 URL 地址栏中查询字符串内容,如果带参数,则修改地址栏中的查询字符串内容,修改后返回一个 \$location 对象
- path : 不带参数,返回页面当前的路径,如果带参数,则修改页面当前的路径,修改后返回一个 \$location 对象
- replace : 取消浏览器导航条中 后退按钮,一般用于表单中提交数据后



- path : 不带参数,返回页面当前的路径,如果带参数,则修改页面当前的路径,修改后返回一个 \$location 对象
- 使用 path 与 \$anchorScroll 服务结合可以根据 id 滚动到对应的位置
- replace : 取消浏览器导航条中 后退按钮,一般用于表单中提交数据后

- 当通过 \$location 对象修改 URL 地址时,将会触发多个 \$location 内置的时间
- 改变地址前,将触发 \$locationChangeStart 事件,该事件可以被 preventDefault 方法阻止
- 地址成功修改后,将触发 \$locationChangeSuccess 事件
- 通过绑定事件,可以侦测用户的登陆和授权的状态,并根据这些状态进行下一步的操作

- 在这个事件中, \$location 服务将开始加载 URL 变化时各种所需的依赖,该事件在触发时带有3个参数
- evt 对象 : 表示触发时原始的事件对象
- current : 表示当前的 URL 地址
- previous : 表示上一级的 URL 地址,如果是首次加载,那么该值为 undefined

- 在 \$locationChangeSuccess事件触发之前,一定触发了 \$locationChangeStart 事件,但触发了 locationChangeStart 事件,不一定会触发 \$locationChangeSuccess事件,因为这个事件同样也可以通过调用 event.preventDefault 方法进行取消
- 在 \$locationChangeSuccess 事件中,也有3个参数,表示的含义与 \$locationChangeStart 事件一致
- 可以使用 \$scope 的\$on 方法进行事件的监听

## 二、其他服务

1.\$cacheFactory

2.\$timeout、 \$interval

3.\$sce

- 缓存是 web 开发中一个非常重要的概念,它的核心组成是一个个键 key / 值 value 存储集合,其中,一个键对应一块缓存内容
- 如果需要请求缓存内容时,发现对应的键存在且有效,才直接通过键返回对应的缓存内容;如果发现对应的键不存在或无效,则重新获取原始数据,并根据键名将对应的缓存内容写入存储集合中,以备下次使用
- 缓存的功能就是加载获取内容的速度,减少重复请求
- 在 Angular 中,可以使用 \$cacheFactory 来生成缓存对象

- \$cacheFactory(key) 表示创建缓存对象,key 代表缓存的 id
- Info() : 返回缓存对象的一些信息,包括大小,名称
- Put() : 向缓存对象中以 key/value 的形式添加缓存内容
- Get() : 可以获取键名对应的键值内容
- Remove() : 可以移除指定键名的缓存
- removeAll() : 用于移除全部的缓存内容

- `$timeout` : Angular 中封装的超时调用,有两个参数,第一个参数为执行的函数,第二个参数为超时调用的时间,执行函数只执行一次
- `$interval` : Angular 中封装的间歇调用,有两个参数,第一个参数为执行的函数,第二个参数为间隔的时间,执行函数会多次调用
- 通过 `$interval.cancel(时间标记)` 取消间歇调用



- 前面学习 ng-bind-html 的时候,了解其可以解析 html 代码,但是必须得导入 sanitize.js 文件,并注入 ngSanitize 依赖,使用起来比较麻烦
- 可以使用 \$sce 的 trustAsHtml() 方法,将 html 代码作为方法的形参就可以直接解析

## 三、路由

1. 路由的介绍
2. 路由的使用

- 可以实现多视图的单页Web应用(single page web application , SPA)
- 允许通过不同的 URL 访问不同的内容
- 当点击以下的任意一个链接时，向服务端请的地址都是一样的  
(<http://www.wuxianedu.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

```
'http://www.wuxianedu.com/#/first'  
'http://www.wuxianedu.com/#/second'  
'http://www.wuxianedu.com/#/third'
```

- 引入 route.js 文件
- 在模块声明中注入对 ngRoute 的依赖
- 在 config 函数中注入 \$routeProvider,对路由进行配置(定义路由表)
- 在主模板中指定加载子视图的位置(配合 ng-view 使用)

- `$routeProvider`提供了定义路由表的服务，它有两个核心方法，`when(path,route)`和`otherwise(params)`
- `when(path,route)`方法接收两个参数，`path`是一个string类型，表示该条路由规则所匹配的路径，它将与地址栏的内容( `$location.path`) 值进行匹配。如果需要匹配参数，可以在`path`中使用冒号加名称的方式，如：  
`path`为`/show/:name`，如果地址栏是`/show/tom`，那么参 数`name`和所对应的值`tom`便会被保存在`$routeParams`中，像这样：`{name : tom}`。  
我们也可以用`*`进行模糊匹配，如：`/show*/:name`将匹配  
`/showInfo/tom`。
- 可以使用 `$routeParams` 服务获取到匹配的参数

- route 是一个object,用来指定当 path 匹配后所需的一系列配置项,该对象有以下属性
- controller : 在当前模板上执行的controller函数
- template : 视图所用的模板,这部分内容将被ngView引用
- templateUrl : 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
- redirectTo : 重定向地址

- Otherwise(parms) 方法是对应路径找不到时的情况,可以配置一个 redirectTo 参数,让其重定向至404页面或者首页
- 单页面程序都是局部刷新的,这个局部指的就是 ng-view 指令的内容,只需在模板中简单的使用此指令,在哪里用,哪里就是局部

- `$routeChangeStart` : 这个事件会在路由跳转前触发
- `$routeChangeSuccess` : 这个事件在路由跳转成功后触发
- `$routeChangeError` : 这个事件在路由跳转失败后触发



- 在Angular 内部,并不支持 \$location 对象提供的方法与页面中的元素进行双向的数据绑定,如果要实现这种效果,需要将对象的方法定义为元素的 model 属性,专门用于绑定 model 属性,并添加两个 watch 方法,当 \$location 的更新时,分别向元素和控制器相互推送更新信息

## 四、ngAnimate

1. ngAnimate 的介绍
2. ngAnimate 的使用
3. ngAnimate 的使用方式

- AngularJS 提供了动画效果，可以配合 CSS 使用。
- ngAnimate 模型可以添加或移除 class 。
- ngAnimate 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 使用动画需要引入 angular-animate.min.js 库
- 还需在创建模块的时候依赖注入 ngAnimate
- 三种方式使用 ngAnimate

- 预定义的类
  - ng-enter
  - ng-enter-active
  - ng-leave
  - ng-leave-active
- 支持的指令
  - ng-if,ng-view,ng-repeat
  - include,switch

- 预定义的类
  - ng-hide-add
  - ng-hide-add-active
  - ng-hide-remove
  - ng-hide-remove-active
- 支持的指令
  - ng-class,ng-show
  - ng-hide,ng-model

- 使用 animation() 方法创建动画
- Animation(选择器,fn) 第一个参数以字符串的方式传入选择器,第二个参数是动画的执行函数,通过函数的返回值设置动画具体要执行的内容
- 使用方式与 CSS3 方式类似

```
app1.animation('.box',function () {  
    return {  
        enter: function (element,done) {  
            $(element).css({width: 0,height: 0});  
            $(element).animate({width: 200,height: 200},1000,done);  
        },  
        leave: function (element,done) {  
            $(element).animate({width:0,height:0},1000,done);  
        }  
    }  
});
```

```
app1.animation('.box',function () {  
  return {  
    addClass:function(element,Sclass,done){  
      console.log(Sclass);  
      $(element).animate({width:0,height:0},1000,done);  
    },  
    removeClass:function(element,Sclass,done){  
      $(element).css({width:0,height:0});  
      $(element).animate({width:200,height:200},1000,done);  
    }  
  }  
});
```

指令名字



- 进入链接时,链接 # 号之后的内容在向服务端请求时会被浏览器忽略掉
- 使用路由的时候需要导入 route.js 文件,再依赖注入 ngRoute
- 路由要搭配 ng-view 才能显示跳转的内容
- 可以使用 \$routeParams 服务获取到链接里面匹配的参数
- 在 Angular 中,使用 ngAnimate 可以为元素添加动画,也可以配合 CSS 使用



慧科集团旗下企业



官方微信

**未经允许不得将视频用于商业用途，否则将追究其法律责任！**

**官方网站：** <http://www.wuxianedu.com/>

**无限智造 互联精英**

**版权所有：无限互联**