



ui-router - 路由控制\$stateProvider

收藏人: KILLKISS

转发到随笔

2015-05-17 | 阅: 7581 转: 25 | 来源   | 分享 

在你的应用中大多数状态都有与其相关联的 url, 路由控制不是设计完成 state 之后的事后想法, 而是开始开发时就应该考虑的问题。

这里是如何设置一个基本url。

```
1 $stateProvider
2   .state('contacts', {
3     url: "/contacts",
4     templateUrl: 'contacts.html'
5   })
```

当我们访问index.html/contacts时, 'contacts'状态将被激活, 同时index.html中的ui-view将被'contacts.html'填充。或者, 通过transitionTo('contacts')方法将状态转变到'contacts'状态, 同时 url 将更新为index.html/contacts。

URL参数

基本参数

通常, url动态部分被称为参数, 有几个选项用于指定参数。基本参数如下:

```
1 $stateProvider
2   .state('contacts.detail', {
3     // 这里设置了url参数
4     url: "/contacts/:contactId",
5     templateUrl: 'contacts.detail.html',
6     controller: function ($stateParams) {
7       // If we got here from a url of /contacts/42
8       expect($stateParams).toBe({contactId: 42});
9     }
10  })
```

或者, 你也可以使用花括号的方式来指定参数:

```
1 // 与前面的设置方法等效
2 url: "/contacts/{contactId}"
```

示例:

'/hello/' - 只匹配'/hello/'路径, 没有对斜杠进行特殊处理, 这种模式将匹配整个路径, 而不仅仅是一个前缀。
'/user/:id' - 匹配'/user/bob'、'/user/1234!!!', 甚至还匹配 '/user/', 但是不匹配 '/user'和'/user/bob/details'。第二个路径段将被捕获作为参数"id"。
'/user/{id}' - 与前面的示例相同,但使用花括号语法。

含正则表达式的参数

使用花括号的方式可以设置一个正则表达式规则的参数:

```
1 // 只会匹配 contactId 为1到8位的数字
2 url: "/contacts/{contactId:[0-9]{1,8}}"
```

示例：

```
'/user/{id:[^/]*}' - 与'/user/{id}'相同  
'/user/{id:[0-9a-fA-F]{1,8}}' - 与前面的示例相似，但只匹配1到8为的数字和字符  
'/files/{path:.*)}' - 匹配任何以'/files/'开始的URL路径，并且捕获剩余路径到参数'path'中。  
'/files/*path' - 与前面相同，捕获所有特殊的语法。
```

警告：不要把捕获圆括号写进正则表达式，ui-router 的 UrlMatcher 将为整个正则表达式添加捕获。

Query Parameters

可以通过?来指定参数作为查询参数

```
1 url: "/contacts?myParam"  
2 // 匹配 "/contacts?myParam=value"
```

如果你需要不止一个查询参数，请用&分隔：

```
1 url: "/contacts?myParam1&myParam2"  
2 // 匹配 "/contacts?myParam1=value1&myParam2=wowcool"
```

嵌套状态的路由控制

附加的方式（默认）

在嵌套状态的路由控制中，默认方式是子状态的 url 附加到父状态的 url 之后。

```
1 $stateProvider  
2   .state('contacts', {  
3     url: '/contacts',  
4     ...  
5   })  
6   .state('contacts.list', {  
7     url: '/list',  
8     ...  
9   });
```

路由将成为：

'contacts' 状态将匹配"/contacts"
'contacts.list' 状态将匹配"/contacts/list"。子状态的url是附在父状态的url之后的。

绝对路由 (^)

如果你使用绝对 url 匹配的方式，那么你需要给你的url字符串加上特殊符号"^"。

```
1 $stateProvider  
2   .state('contacts', {  
3     url: '/contacts',  
4     ...  
5   })  
6   .state('contacts.list', {  
7     url: '^/list',  
8     ...  
9   });
```

路由将成为：

```
'contacts' 状态将匹配"/contacts"
'contacts.list' 状态将匹配"/list"。子状态的url没有附在父状态的url之后的，因为使用了^。
```

\$stateProvider 服务

之前看到的\$stateProvider服务是一个对象，包含 url 中每个参数的键/值。\$stateProvider可以为控制器或者服务提供 url 的各个部分。

注意：\$stateProvider服务必须与一个控制器相关，并且\$stateProvider中的“键/值”也必须事先在那个控制器的url属性中有定义。

```
1 // 如果状态中 url 属性是：
2 url: '/users/:id/details/{type}/{repeat:[0-9]+}?from&to'
3
4 // 当浏览
5 '/users/123/details//0'
6
7 // $stateProvider 对象将是
8 { id:'123', type:'', repeat:'0' }
9
10 // 当浏览
11 '/users/123/details/default/0?from=there&to=here'
12
13 // $stateProvider 对象将是
14 { id:'123', type:'default', repeat:'0', from:'there', to:'here' }
```

使用\$stateProvider的两个陷阱

只有当状态被激活并且状态的所有依赖项都被注入时，\$stateProvider对象才存在。这代表你不能再状态的resolve函数中使用\$stateProvider对象，可以使用\$state.current.params来代替。

```
1 $stateProvider.state('contacts.detail', {
2   resolve: {
3     someResolve: function($state){
4       /*** 不能在这里使用 $stateProvider , the service is not ready ***/
5       /*** 使用 $state.current.params 来代替 ***/
6       return $state.current.params.contactId + "!"
7     };
8   },
9   // ...
10 })
```

在状态控制器中，\$stateProvider对象只包含那些在状态中定义的参数，因此你不能访问在其他状态或者祖先状态中定义的参数。

```
1 $stateProvider.state('contacts.detail', {
2   url: '/contacts/:contactId',
3   controller: function($stateProvider){
4     $stateProvider.contactId /*** Exists! ***/
5   }
6 }).state('contacts.detail.subitem', {
7   url: '/item/:itemId',
8   controller: function($stateProvider){
9     $stateProvider.contactId /*** 注意! DOESN'T EXIST!! ***/
10    $stateProvider.itemId /*** Exists! ***/
11  }
12 })
```

\$urlRouterProvider

\$urlRouterProvider负责处理在状态配置中指定的url路由方式之外的 url 请求的路由方式。\$urlRouterProvider负责监视\$location，

当\$location改变后，\$urlRouterProvider将从一个列表，一个接一个查找匹配项，直到找到。所有 url 都编译成一个UrlMatcher对象。

\$urlRouterProvider有一些实用的方法，可以在module.config中直接使用。

when() for redirection 重定向

参数：

what **String | RegExp | UrlMatcher**, 你想重定向的传入路径

handler **String | Function** 将要重定向到的路径

handler 作为 **String**

如果handler是字符串，它被视为一个重定向，并且根据匹配语法决定重定向的地址。

```
1 app.config(function($urlRouterProvider){
2   // when there is an empty route, redirect to /index
3   $urlRouterProvider.when('', '/index');
4
5   // You can also use regex for the match parameter
6   $urlRouterProvider.when('/aspx/i', '/index');
7 })
```

handler 作为 **Function**

如果handler是一个函数，这个函数是注入一些服务的。如果\$location匹配成功，函数将被调用。你可以选择性注入\$match。

函数可以返回：

falsey 表明规则不匹配，\$urlRouter将试图寻找另一个匹配

String 该字符串作为重定向地址并且作为参数传递给\$location.url()

nothing或者任何为真的值，告诉\$urlRouterurl 已经被处理

示例：

```
1 $urlRouterProvider.when(state.url, ['$match', '$stateParams', function ($match, $stateParams) {
2   if ($state.$current.navigable != state || !equalForKeys($match, $stateParams)) {
3     $state.transitionTo(state, $match, false);
4   }
5 }]);
```

otherwise() 无效路由

参数：

path **String | Function** 你想重定向url路径或者一个函数返回url路径。函数可以包含\$injector和\$location两个参数。

```
1 app.config(function($urlRouterProvider){
2   // 在配置（状态配置和when()方法）中没有找到url的任何匹配
3   // otherwise will take care of routing the user to the specified url
4   $urlRouterProvider.otherwise('/index');
5
6   // Example of using function rule as param
7   $urlRouterProvider.otherwise(function($injector, $location){
8     ... some advanced code...
9   });
10 })
```

rule() 自定义url处理

参数：

handler **Function** 一个函数，包含\$injector和\$location两个服务作为参数，函数负责返回一个有效的路径的字符串。

```
1 app.config(function($urlRouterProvider){
2     // Here's an example of how you might allow case insensitive urls
3     $urlRouterProvider.rule(function ($injector, $location) {
4         var path = $location.path(), normalized = path.toLowerCase();
5         if (path !== normalized) return normalized;
6     });
7 })
```

\$urlMatcherFactory 和 UrlMatchers

定义了url模式和参数占位符的语法。\$urlMatcherFactory是在幕后被\$urlRouterProvider调用，来缓存编译后的UrlMatcher对象，而不必在每次 location 改变后重新解析url。大多数用户不需要直接使用\$urlMatcherFactory方法，但是在状态配置中非常实用，可以使用\$urlMatcherFactory方法来生成一个UrlMatcher对象，并在状态配置中使用该对象。

```
1 var urlMatcher = $urlMatcherFactory.compile("/home/:id?param1");
2 $stateProvider.state('myState', {
3     url: urlMatcher
4 });
```

AngularJS ui-router

来自: [KILLKISS](#) > [《Angular》](#)

上一篇: [AngularJS开发指南28: 创建服务](#)

下一篇: [angular的post请求, SpringMVC后台接收不到参数值的解决方案](#)

转藏到我的图书馆

 献花(0)

分享到微信

以文找文

分享:



类似文章

[更多](#)

热门推荐

URL传递中文参数的问题

观模式:为子系统中的一组接口提供一个一...

Content Provider 基础 之URI - Android...

flex 强大的navigateToURL方法

jquery 弹出遮罩插件 prettyPhoto 参数说....

Android异步下载网络图片 (其二)

Servlet六个取URL的函数 A A A

每个 Web 开发者都应该知道的关于 URL ...

猜你喜欢

 换一组



感悟"经典", 走好人生路



教你轻松降血脂



治失眠的最好食物, 不可错过



一张图告诉你优秀驾驶员开车技巧



警惕:女性经期千万不要吃这些水果



中医治疗高血压



养生养心100条



小学数学公式大全



人体器官24小时变化规律



老祖宗留下来的养生经验

发表评论:

如果您只想对文章收藏人说声谢谢, 请点击正文下方的“献花”...

