**angular-ui-router 多视图views**

[angular-ui-router API](http://angular-ui.github.io/ui-router/site/#/api)

UI Router 中有三种方式激活一个路由：   
（1）$state.go()：优先级较高的便利方式   
（2）ui-sref：点击包含此指令跳转   
（3）url：url导航

**一、$state.go()**

**（1）$state.go(to [, toParams] [, options])**   
参数：   
to：绝对“state名称”或者相对的“state路径”（如果路径，以“^或者.”是相对，否则为绝对）   
toParams：发送给state的数据参数，由$stateParams构建   
options：{ location: true, inherit: true, relative: $state.$current, notify: true }   
**（2）$state.transitionTo(to, toParams [, options])**   
$state.go() 内部调用此方法   
**（3）$state.reload()**   
**（4）$state.includes(stateName [, params]) stateName是否为当前路由的一部分**   
**（5）$state.is(stateOrName [, params]) stateOrName是否为当前路由（完全匹配）**   
注意：contact.details.item === contactDetailsItem   
**（6）$state.href(stateOrName [, params] [, options])**   
**（7）$state.get([stateName])**   
**（8）$state.current**

参考地址：[https://github.com/angular-ui/ui-router/wiki/Quick-Reference#stategoto–toparams–options](https://github.com/angular-ui/ui-router/wiki/Quick-Reference" \l "stategoto--toparams--options" \t "_blank)

**二、ui-sref**

此指令必须绑定到**<a>**标签，如果该路由有对应的关联URL，其通过$state.href()自动生成和更新href属性。

<a ui-sref="home">Home</a> | <a ui-sref="about">About</a>

* 1

（1）ui-sref-active=”active” 该路由激活，则对应增加active样式名称

<ul>

<li ui-sref-active="active" class="item">

<a href ui-sref="app.user({user: 'bilbobaggins'})">@bilbobaggins</a>

</li>

</ul>

当路由是“pp.user”，且包含参数user值为’bilbobaggins’，会变成如下

<ul>

<li ui-sref-active="active" class="item active">

<a ui-sref="app.user({user: 'bilbobaggins'})" href="/users/bilbobaggins">@bilbobaggins</a>

</li>

</ul>

（2）ui-sref-opts 传递参数

<a ui-sref="home" ui-sref-opts="{reload: true}">Home</a>

参考地址：[https://github.com/angular-ui/ui-router/wiki/Quick-Reference#ui-sref](https://github.com/angular-ui/ui-router/wiki/Quick-Reference" \l "ui-sref" \t "_blank)

**三、URL Routing**

$stateProvider

.state('contacts', {

url: "/contacts",

templateUrl: 'contacts.html'

})

其支持正则、查询参数

**（1）$urlRouterProvider.when()**   
**（2）$urlRouterProvider.otherwse()**   
**（3）$urlRouterProvider.rule()**

var myApp = angular.module("myApp", ["ui.router"]);

myApp.config(function ($stateProvider, $urlRouterProvider) {

$urlRouterProvider.when('/', '/index');

$urlRouterProvider.otherwise('/index');

});

参考地址：[https://github.com/angular-ui/ui-router/wiki/URL-Routing](https://github.com/angular-ui/ui-router/wiki/URL-Routing" \t "_blank)

**四、示例（多视图：页面可以显示多个动态变化的不同区块）**

// Node静态服务

var http = require("http");

var express = require("express");

var serveStatic = require('serve-static');

var app = new express();

app.use(serveStatic(\_\_dirname + '/'));

app.get("/", function(req, res) {

res.sendFile(\_\_dirname + "/Main2.html");

})

app.listen(1337, "localhost");

<!-- Main2.html -->

<!DOCTYPE html>

<head>

<title></title>

<script src="./angular.js"></script>

<script src="./angular-ui-router.js"></script>

<script src="./App2.js"></script>

</head>

<body data-ng-app="myApp">

<h1>多ui-view</h1>

<div ui-view></div>

<div ui-view="chart"></div>

<div ui-view="data"></div>

</body>

<html>

// App2.js

var myApp = angular.module("myApp", ["ui.router"]);

myApp.config(function ($stateProvider, $urlRouterProvider) {

$urlRouterProvider.when("", "/home");

$stateProvider.state("home", {

url: "/home",

views: {

"": {

template: "<h1>HELLO!</h1>"

},

"chart": {

template: "chart"

},

"data": {

template: "data"

}

}

})

.state("index", {

url: "/index",

views: {

"": {

template: "<h1>HELLO!</h1>"

},

"data": {

template: "data-index"

}

}

})

});

**五、示例（嵌套视图：页面某个动态变化区块中，嵌套着另一个可以动态变化的区块）**

// Node静态服务

var http = require("http");

var express = require("express");

var serveStatic = require('serve-static');

var app = new express();

app.use(serveStatic(\_\_dirname + '/'));

app.get("/", function(req, res) {

res.sendFile(\_\_dirname + "/Main.html");

})

app.listen(1337, "localhost");

<!-- Main.html -->

<!DOCTYPE html>

<head>

<title></title>

<script src="./angular.js"></script>

<script src="./angular-ui-router.js"></script>

<script src="./App.js"></script>

</head>

<body data-ng-app="myApp">

<h1>AngularJS Home Page (Ui-router Demonstration)</h1>

<div ui-view></div>

</body>

<html>

// App.js

var myApp = angular.module("myApp", ["ui.router"]);

myApp.config(function ($stateProvider, $urlRouterProvider) {

$urlRouterProvider.when("", "/PageTab");

$stateProvider

.state("PageTab", {

url: "/PageTab",

templateUrl: "PageTab.html"

})

.state("PageTab.Page1", {

url:"/Page1",

templateUrl: "Page1.html"

})

.state("PageTab.Page2", {

url:"/Page2",

templateUrl: "Page2.html"

})

.state("PageTab.Page3", {

url:"/Page3",

templateUrl: "Page3.html"

});

});

<!-- PageTab.html -->

<div>

<div>

<span style="width:100px" ui-sref=".Page1"><a href="">Page-1</a></span>

<span style="width:100px" ui-sref=".Page2"><a href="">Page-2</a></span>

<span style="width:100px" ui-sref=".Page3"><a href="">Page-3</a></span>

</div>

<div>

<div ui-view/>

</div>

</div>

<!-- Page1.html -->

<div>

<div>

<h1>Page 1 content goes here...</h1>

</div>

</div>

<!-- Page2.html -->

<div>

<div>

<h1>Page 2 content goes here...</h1>

</div>

</div>

<!-- Page3.html -->

<div>

<div>

<h1>Page 3 content goes here...</h1>

</div>

</div>