**ionic ion-tap选项卡以及 路由结合ion-tap详解**

学习要点：

1. 选项卡 : ion-tabs简介
2. ion-tabs 常用设置
3. ion-tabs 下面的 ion-tab ：标题文字、图标和徽章 显示隐藏
4. ion-tabs 事件 和 $ionicTabsDelegate
5. ion-tabs路由详解

**主讲教师：（树根）**

**合作网站：[www.phonegap100.com](http://www.phonegap100.com(PhoneGap)** **(PhoneGap中文网)**

**合作网站：[www.itying.com](http://www.itying.com) (IT营)**

1. **选项卡 : ion-tabs简单介绍**

使用ion-tabs指令声明选项卡，使用ion-tab声明选项页：

<ion-tabs>

<ion-tab title="xxx">...</ion-tab>

<ion-tab title=" xxx ">...</ion-tab>

...

</ion-tabs>

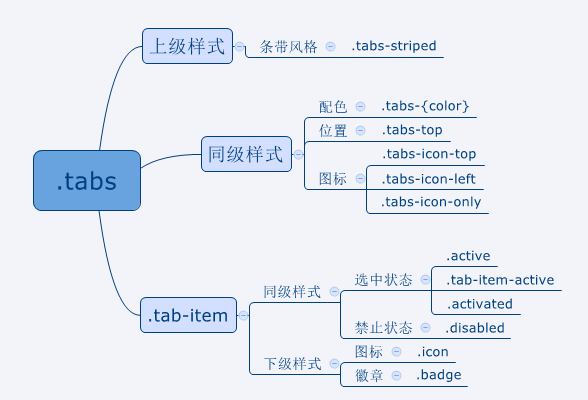
每个ion-tab元素的title属性值将作为选项页的标题,其内容将填充选项卡书签栏之外的剩余区域（被应用.pane样式）。

**注意：**

1. 不要把ion-tabs指令放在ion-content之内

2. ion-tab的内容应当放入ion-view指令内，否则ionic在计算布局时可能出错

AngularJS编译后，ion-tabs元素将被应用.tabs样式，因此我们可以使用 相关的样式调整ion-tabs的外观：



|  |
| --- |
| <ion-tabs>  <ion-tab title="tab1">  <ion-view>  <ion-content class="calm-bg">  tab 1 content  </ion-content>  </ion-view>  </ion-tab>  <ion-tab title="tab2">  <ion-view>  <ion-content class="balanced-bg">  tab 2 content  </ion-content>  </ion-view>  </ion-tab>  <ion-tab title="tab3">  <ion-view>  <ion-content class="energized-bg">  tab 3 content  </ion-content>  </ion-view>  </ion-tab>  </ion-tabs> |

1. **ion-tabs 常用设置**

**ion-tabs声明条带风格：**

如果学习过课程：ionic之CSS框架，应该记得使用**.tabs-striped样式可以将选项卡 声明为条带风格**：

**.tabs-standard 申明不带条风格**

<ion-tabs class="tabs-striped">...</ion-tabs>

也可以通过$ionicConfigProvider在AngularJS的配置阶段，将选项卡设置为条带风格：

app.config(function($ionicConfigProvider){

$ionicConfigProvider.tabs.style("striped"); // 参数可以是： standard | striped

})

**ion-tabs : 声明位置：**

如果学习过课程：ionic之CSS框架，应该记得使用.tabs-top可以将选项卡置于 顶部标题栏之下：

<ion-tabs class="tabs-top">...</ion-tabs>

也可以通过$ionicConfigProvider，在配置阶段设置选项卡的位置是在顶部还是底部：

app.config(function($ionicConfigProvider){

$ionicConfigProvider.tabs.position("top"); //参数可以是：top | bottom

});

1. **ion-tabs 下面的 ion-tab ： 标题文字、图标和徽章显示隐藏**

ion-tab指令有以下属性用于设置文本、图标和徽章：

**title - 标题文字**

标题文字是必须的。该属性值将作为选项页的标题文字。

**icon - 标题图标**

使用icon属性是可选的，该属性值将用来在标题文字旁边添加一个指定的图标。 这个属性的值将被作为icon-on和icon-off的默认值

**icon-on - 被选中状态的标题图标**

如果一个选项页被选中，ion-tabs将使用icon-on属性的值绘制图标。如果icon-on 没有设置，那么ion-tabs就使用icon属性的值绘制图标

**icon-off - 未选中状态的标题图标**

如果一个选项页没有被选中，ion-tabs将使用icon-off属性的值绘制图标。如果icon-off 没有设置，那么ion-tabs就使用icon属性的值绘制图标

**badge - 标题徽章**

ion-tabs使用badge属性的值在标题文字旁边添加一个圆形的文字标识，通常用来 显示数字。这个属性是可选的，可以是一个具体的值，也可以是当前作用域上的 一个变量

**badge-style - 标题徽章样式**

使用badge-style属性设置徽章的样式， 比如配色方案：barge-{color}

**hidden - 隐藏**

hidden属性是当前作用域上的表达式。 当其值为true时，选项页将不可见

**disabled - 禁止**

disabled属性是当前作用域上的表达式。当值为true时，选项页将不响应 用户的点击

|  |
| --- |
| <ion-tabs class="tabs-positive tabs-icon-top">  <ion-tab title="Home" icon-on="ion-ios-filing" icon-off="ion-ios-filing-outline">  <ion-view>  <ion-header-bar class="bar-positive">  <h1 class="title">home tab</h1>  </ion-header-bar>  <ion-content>  <p>home content</p>  </ion-content>  </ion-view>  </ion-tab>  <ion-tab title="About" icon-on="ion-ios-clock" icon-off="ion-ios-clock-outline" badge="12" badge-style="badge-assertive">  <ion-view>  <ion-header-bar class="bar-positive">  <h1 class="title">about tab</h1>  </ion-header-bar>  <ion-content>  <p>about content</p>  </ion-content>  </ion-view>  </ion-tab>  <ion-tab title="Settings" icon-on="ion-ios-gear" icon-off="ion-ios-gear-outline">  <ion-view>  <ion-header-bar class="bar-positive">  <h1 class="title">settings tab</h1>  </ion-header-bar>  <ion-content>  <p>settings content</p>  </ion-content>  </ion-view>  </ion-tab>  </ion-tabs> |

1. **ion-tabs 事件 和 $ionicTabsDelegate**

**1.ion-tab事件：**

**on-select - 选中事件**

可选。选项页从未选中状态切换到选中状态时执行此表达式

**on-deselect - 未选中事件**

可选。选项页从选中状态切换到未选中状态时执行此表达式

**ng-click - 点击事件**

可选。选项页被点击时执行此表达式。如果这个属性被设置，那么ion-tabs将不会 自动切换选项页，调用者需要手动调用$ionicTabsDelegate的select()方法进行 选项页切换

**2.ionicTabsDelegate**

使用**$ionicTabsDelegate**服务，我们可以在脚本中控制选项卡对象：

select(index) - 选中指定的选项页

index参数从0开始，第一个选项页的index为0，第二个为1，依次类推。

selectedIndex() - 返回当前选中选项页的索引号

如果当前没有选中的选项页，则返回 -1。

|  |
| --- |
| angular.module("ezApp",["ionic"])  .controller("ezCtrl",function($scope,$ionicTabsDelegate,$interval){  var idx=0;  $interval(function(){  idx = (idx + 1) % 3;  $ionicTabsDelegate.select(idx);  },2000);  }); |

1. **ion-tabs路由详解**
2. **触发状态迁移的几种方式（通俗的讲就是页面跳转的几种方式）**

1. 调用$state.go() 方法，这是一个高级的便利方法；  
2. 点击包含 ui-sref 指令的链接 <a ui-sref="state1">Go State 1</a>  
3. 导航到与状态相关联的 url。

1. **通过href方式页面切换需要指定 url**

|  |
| --- |
| var app = angular.module("myApp", ["ionic"]);  app.config(function($stateProvider,$urlRouterProvider) {  $stateProvider  .state("home", {  url: "/home",  templateUrl: "home.html",  controller:'homeController'  })  .state("music", {  **url**: "/music",  templateUrl: "music.html",  **controller**:'musicController'  });  })  .controller("myCtrl",function($scope,$state){  $state.go("home");  })  //  .controller("homeController",function($scope){  console.log('home');  })  .controller("musicController",function($scope){  console.log('music');  })  </script> |

1. **ionic中结合tab状态嵌套的几种方式**

**状态可以相互嵌套。有三个嵌套的方法：**

1.使用“点标记法”，例如：.state('contacts.list', {})

2.使用parent属性，指定一个父状态的名称字符串，例如：parent: 'contacts'

3.使用parent属性，指定一个父状态对象，例如：parent: contacts（contacts 是一个状态对象）

**1.点标记法**

在$stateProvider中可以使用点语法来表示层次结构，下面，contacts.list是contacts的子状态。

|  |
| --- |
| $stateProvider  .state('contacts', {})  .state('contacts.list', {}); |

**2.使用parent属性，指定一个父状态的名称字符串**

|  |
| --- |
| $stateProvider  .state('contacts', {})  .state('list', {  parent: 'contacts'  }); |

**3.基于对象的状态**

如果你不喜欢使用基于字符串的状态，您还可以使用基于对象的状态。name属性将在状态对象内部设置，在所有的子状态对象中设置parent属性为父状态对象，像下面这样：

|  |
| --- |
| var contacts = {  name: 'contacts', //mandatory  templateUrl: 'contacts.html'  }  var contactsList = {  name: 'list', //mandatory  parent: contacts, //mandatory  templateUrl: 'contacts.list.html'  }  $stateProvider  .state(contacts)  .state(contactsList) |

1. **ionic路由结合tap实现页面切换**
2. **在ionic tab中定义ion-nav-view 并且加上name属性**

<ion-nav-view name="news-list"></ion-nav-view>

|  |
| --- |
| <ion-tabs class="tabs-icon-top tabs-positive">  <!-- Dashboard Tab -->  <ion-tab title="list" icon-off="ion-ios-pulse" icon-on="ion-ios-pulse-strong" href="#/news/list">  <ion-nav-view name="news-list"></ion-nav-view>  </ion-tab>  <!-- Chats Tab -->  <ion-tab title="content" icon-off="ion-ios-chatboxes-outline" icon-on="ion-ios-chatboxes" href="#/news/content">  <ion-nav-view name="news-content"></ion-nav-view>  </ion-tab>  </ion-tabs> |

1. **在ionic $stateProvider.state中定义view 并对应 ion-nav-view 中的name属性**

|  |
| --- |
| $stateProvider.state('news', {  url: "/news",  abstract:true,  templateUrl: "templates/news.html"  })  .state('news.list', {  url: '/list',  views: {  '**news-lis**t': {  templateUrl: 'templates/news-list.html'  }  }  }) |

1. **ionic** **states 抽象状态abstract**

一个抽象的状态可以有子状态但不能显式激活，它将被隐性激活当其子状态被激活时。

**下面是两个最常用的抽象状态的示例：**

•为所有子状态预提供一个基url

•在父状态中设置template属性，子状态对应的模板将插入到父状态模板中的ui-view(s)中