

阿东入门系列《二》轮播，Tab 分页

一：Tab 分页

1.1 组件 tabBar

微信小程序提供了一个组件 tabBar 用来实现多 tab 分页功能。如果我们的程序是一个多 tab 应用（客户端窗口的底部有 tab 栏可以切换页面），那么我们可以通过 tabBar 配置项指定 tab 栏的表现，以及 tab 切换时显示的对应页面。

tabBar 是一个数组，只能配置最少 2 个、最多 5 个 tab，tab 按数组的顺序排序。

属性说明：

属性	类型	必填	默认值	描述
color	HexColor	是		tab 上的文字默认颜色
selectedColor	HexColor	是		tab 上的文字选中时的颜色
backgroundColor	HexColor	是		tab 的背景色
borderStyle	String	否	black	tabbar上边框的颜色， 仅支持 black/white
list	Array	是		tab 的列表，详见 list 属性说明，最少2个、最多5个 tab
position	String	否	bottom	可选值 bottom、top

其中 list 接受一个数组，数组中的每个项都是一个对象，其属性值如下：

属性	类型	必填	说明
pagePath	String	是	页面路径，必须在 pages 中先定义
text	String	是	tab 上按钮文字
iconPath	String	是	图片路径，icon 大小限制为40kb

微信小程序提供了一个组件 tabBar 用来实现多 tab 分页功能。如果我们的程序是一个多 tab 应用（客户端窗口的底部有 tab 栏可以切换页面），那么我们可以通过 tabBar 配置项指定 tab 栏的表现，以及 tab 切换时显示的对应页面。

1.2 配置位置

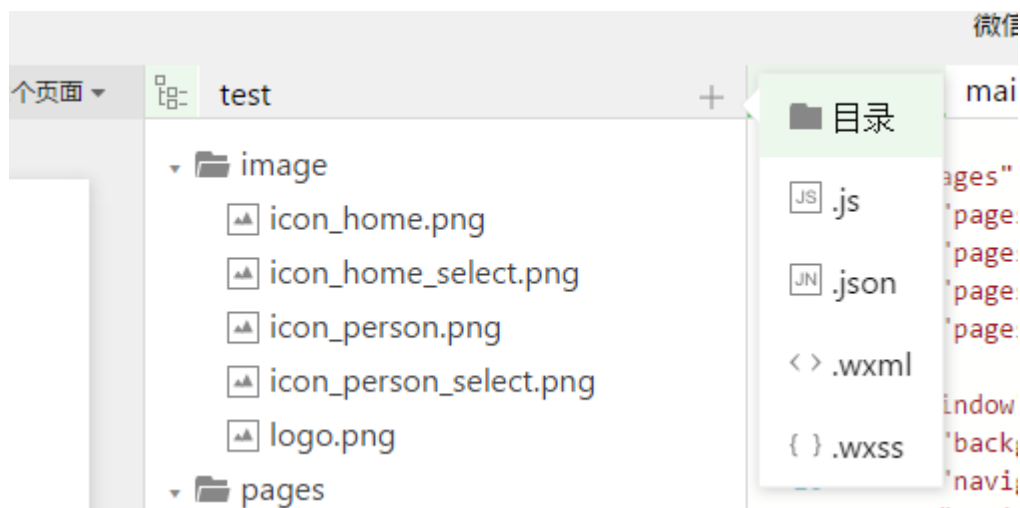
微信小程序中认为整个页面分为 导航栏 navigationBar，tab 栏 tabBar 和 页面区域（就是在 pages 中配置的页面显示区域），所以 tabBar 的配置位于全局配置文件 **app.json** 文件中。

```
"tabBar": {  
  "color": "#a9a9a9",  
  "selectedColor": "#1f1f1f",  
  "backgroundColor": "#fefefe",  
  "borderStyle": "white",  
  "list": [  
    {  
      "pagePath": "pages/main/main",  
      "text": "首页",  
      "iconPath": "image/icon_home.png",  
      "selectedIconPath": "image/icon_home_select.png"  
    },  
  ],  
}
```

```
{  
  "pagePath": "pages/aboutUs/aboutUs",  
  "text": "关于我们",  
  "iconPath": "image/icon_person.png",  
  "selectedIconPath": "image/icon_person_select.png"  
}  
] }
```

1.3 icon 图标

对于每一个 tab 的 icon，需要放置于本地工程内，经测试此处不支持网络图片。需在工程根目录下创建用于存放图片的文件夹，文件名随意，使用时需指定其相对路径。（作者实际操作过程中，没有在 IDE 中找到在 image 文件夹中放入图片文件的方法，我是在本地文件中操作的，不知道是功能太隐蔽还是我没有找到，如果有知道的望留言告知）



```
{  
  "pagePath": "pages/main/main",  
  "text": "首页",  
  "iconPath": "image/icon_home.png",  
  "selectedIconPath": "image/icon_home_select.png"  
}
```

二：轮播

组件 swiper

微信小程序原生就提供了轮播控件，在微信的文档中叫做 滑块视图容器。只需要简单的配置就可以实现轮播的效果。

属性名	类型	默认值	说明
indicator-dots	Boolean	false	是否显示面板指示点
autoplay	Boolean	false	是否自动切换
current	Number	0	当前所在页面的 index
interval	Number	5000	自动切换时间间隔
duration	Number	1000	滑动动画时长
bindchange	EventHandle		current 改变时会触发 change 事件，event.detail = {current: current}

上图是微信开发文档中给出的，但是笔者发现还有一个属性 **vertical**，其值接受 Boolean 型变量，默认值为 false。

当不设置 **vertical** 属性，或者 vertical=" false" 时，指示点在组件下部，图片轮播从左至右，效果如下：



当设置 `vertical="true"` 时，指示点在组件右部，图片轮播从下至上，效果如下：



注意: `swiper` 是一个容器类视图，但是其中只能放置`<swiper-item/>`组件，如放置其他节点，会被自动删除。

swiper-item

仅可放置在 `swiper` 组件中，宽高自动设置为 100%。代表轮播中一帧的页面，通常以循环的方式加载到页面中。

代码如下：

```
<!--main.wxml-->
```

```
<view>
```

```
  <swiper class="swiper_box" indicator-dots="{{indicatorDots}}" vertical="{{vertical}}"
```

```
    autoplay="{{autoplay}}" interval="{{interval}}" duration="{{duration}}" bindchange="swiperchange">
```

```
    <block wx:for="{{images}}">
```

```
      <swiper-item>
```

```
        <image src="{{item.picurl}}" class="slide-image"/>
```

```
      </swiper-item>
```

```
    </block>
```

```
  </swiper>
```

```
</view>
```

```
//main.js
```

```
//获取应用实例
```

```
var app = getApp()
```



```
Page({
  data: {
    indicatorDots: true,

    vertical: true,

    autoplay: true,

    interval: 3000,

    duration: 1000,

    loadingHidden: false // loading
  },
```

//事件处理函数

```
swiperchange: function(e) {

  // 此处写 轮播 改变时会触发的 change 事件
},
```

```
onLoad: function() {

  console.log('onLoad')

  var that = this
```

//sliderList

```
wx.request({
  url: 'http://huanqiuxiaozhen.com/wemall/slider/list',

  method: 'GET',

  data: {},
  header: {

    'Accept': 'application/json'

  },

  success: function(res) {
    that.setData({
      images: res.data
    })
  }
})
})
```

item 单击事件

在 **swiper-item** 上绑定事件，通过 `data` 自定义标签绑定数据。然后在 `function` 中通过 `event` 拿到。

```
<!--main.wxml-->
```

```
<view>
```

```
<swiper class="swiper_box" indicator-dots="{{indicatorDots}}" vertical="{{vertical}}"
```

```
        autoplay="{{autoplay}}" interval="{{interval}}" duration="{{duration}}" bindchange="swiperchange">

<block wx:for="{{images}}">

    <swiper-item bindtap="itemclick" data-id="{{item.id}}" data-name="{{item.name}}">

        <image src="{{item.picurl}}" class="slide-image"/>

    </swiper-item>

</block>

</swiper>

</view>
```

注意代码中的 **data-id** 和 **data-name** 均为自定义标签，然后可以在绑定事件的事件中通过 **id** 和 **name** 拿到。

// 轮播 item 点击事件

```
itemclick: function(e) {
    wx.showToast({
        title: e.currentTarget.dataset.id + "",
        icon: 'success',
        duration: 2000
    })
},
```

注意在绑定的 function 中可以通过 event 拿到对应的数据。如：`e.currentTarget.dataset.id` 对应 wxml 中的 data-id

当然，还有另一种办法。不需要绑定事件，通过在每一个的 swiper-item 外面包上一个 a 标签，以超链接的方式跳转页面。