**weex开发美齐Android应用导航栏的开发**

在使用weex创建Android应用的过程中，选择使用iconfont：<http://www.iconfont.cn> 原因是

1.有很多现成的图标，完全可以满足需求，同时也可对图标进行便捷的修改、下载或在线引用，很是方便。

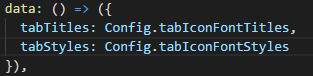
2.同为阿里巴巴的开发工具，容易配合weex，weex-ui使用

在决定底边导航栏的实现方式上，由于weex-ui有现成的方案可使用，故使用weex-ui的wxc-tab-bar：<https://alibaba.github.io/weex-ui/#/cn/packages/wxc-tab-bar/> 组件来实现。

此组件的config单独在一个config.java文件中，并在.vue文件中进行引用，勿忘。

使用iconfont时应当在.vue文件中修改三处位置：





首先需要了解一个概念，即symbol形式的图标：即文字形式的图标，是矢量的，应用文件为文字包文件.ttf

在web端，若wxc-tab-bar组件使用symbol形式的iconfont图标，只要像手册中的示例一样，使用在线生成的symbol链接即可。（不知道图标的编码可以先把自己的iconfont项目下载下来，打开文件就可看到，引用在线url链接时把末尾的.js改成.ttf即可）

当打包成android时，由于weex的原因，引用symbol形式图标是三端不一致的，具体如下：

1. Android或者iOS 访问本地图片或者字体,在weex中统一以' local://'为前缀 ;
2. '/'在android下如果加载的是字体对应的就是assets目录,若果加载的图片就从drawable 目录, 所以iconfont.ttf放置在src/assets目录下的话,字体的url加载方式应该为('local:///iconfont.ttf')

★此为本地离线引用方法！在线引用搞清楚了再补充

所以应当先从iconfont网站中把自己的项目下载并解压，将其中的.ttf文件拷贝到weex工程platforms\android\app\src\main\assets中，然后在config中的url使用'local:///\*\*\*\*.ttf'来引用。

参考网站：<https://segmentfault.com/a/1190000011852209>

初级效果如下



若要根据目前已有app的效果来编写导航栏，则使用weex-ui的wxc-tab-bar组件并不能满足要求，因为其改动空间太小。故重新开始自定义导航栏。

一些点：

1 Weex中的屏幕整体宽度为750px是由官方设定的，据此可以方便的设置各部件的位置，宽度。但高度没有这种情况。

2 class中组件的布局方式用position 绝对布局为 absolute

3 class中的button为距离屏幕底部的距离，设置为0则为在底部

4 weex中长度的默认单位为px

5 weex中组件的布局使用css盒子模型，可参考weex官网wiki中的通用样式

难点一：图标若使用图片，则在点击事件中并不容易切换，故采用了阿里的iconfont字符图标的显示形式，在template中使用

<text class="tab-icon-5" style='font-family:iconfont;font-size:40;'>&#xe6e2;</text>

注意：可以把style中的代码写道该模块的class中

在script中使用（使用export default和module.exports不同）

const domModule = weex.requireModule('dom');

export default{

created () {

domModule.addRule('fontFace', {

'fontFamily': "iconfont",

'src': "url('http://at.alicdn.com/t/font\_668071\_4x4nzm1ahkr442t9.ttf')"

});

}

}

※1 export default中的created()方法类似与android中的onCreate()方法，在创建时被调用，用于一些初始化的操作

2 iconfont使用方式参考<http://weex.apache.org/cn/references/components/text.html>

3 Iconfont代码使用参考<http://dotwe.org/vue/6f2d4314a6fbcb6fee293caa6df991c3>

4 module.exports与exports，export与export default之间的关系和区别

<https://www.cnblogs.com/fayin/p/6831071.html>

5 CommonJS模块规范（module.exports）

<http://javascript.ruanyifeng.com/nodejs/module.html#toc1>

6 ES6模块规范（export default）

<http://es6.ruanyifeng.com/#docs/module#export-default-%E5%91%BD%E4%BB%A4>

难点二：点击事件

采用了类似与vue的v-on点击事件，v-on可用@代替

动作在methods中设置

export default{

methods：{//在此设置函数和方法}

}

※参考weex官网，playground中的example中的语法中的v-on

<http://es6.ruanyifeng.com/#docs/module#export-default-%E5%91%BD%E4%BB%A4>

难点三：点击后图标和文字颜色需要改变

思路是使用两个class属性，但未点击时使用一个class属性，点击后使用第二个class属性。

可以采用weex中的绑定属性值v-bind方法中的类名绑定方法

<text :class="['tab-icon-1', active1 ? 'active': '' ]" @click="page1">&#xe667;</text>

'tab-icon-1'代表第一个class

'active'代表第二个class

active1用于判断使用哪个class，当active1=false时使用前一个class

当active1=true时使用后一个class

使用此方法时需要在export default中先指定active1为false，然后在监听事件方法中设置方法使active1关键字由false变为true从而切换class，如下所示

export default{

data(){

return{

active1:false,

}

},

methods:{

page1(){

this.active1=true;

}

}

※参考参考weex官网，playground中的example中的语法中的v-bind中的类名绑定方法

<http://dotwe.org/vue/5319a828ad7a360e03d556a1a1296d8a>

效果如图

