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:

title-type="iconFont"

```
data: () => ({
  tabTitles: Config.tabIconFontTitles,
  tabStyles: Config.tabIconFontStyles
}),
```

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;'></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

