在 uniapp 中使用 colorui 制作自定义 tabbar

1、直接在 pages. json 定义 tabbar

我们通常在 uniapp 中的 pages.json 中直接配置 tabbar。 如下面的例子:

```
pages.json | 5、使用 colorui制作自定义tabbar pages.json | 自我练习
∨ Ⅲ 5、使用 colorui制作自定义tabbar
> 🖿 .hbuilderx
                                                   "pages": [ //pages数组中第一项表示应用启动页,参考: https://uniapp.dcloud.io/collocation/pages
   > 🖿 colorui
                             1、先在pages更设置好 {
                                                           "path": "pages/home/home",
"style": {
   🗸 🚞 pages 🔫
                                                                                                                 - 3、修改page页,这里用于设置页面
                                                                "navigationBarTitleText": "首页"
      ∨ 📦 home
           M home.vue
     ∨ 🖮 user
                                         "path": "pages/user/user",
"style": {
    "navigationBarTitleText": "用户"
           user.vue
   ✓ 📄 static
                                                home.png
                                                                                                               ■ 4、设置tabbar页
           la home-active.png
     App.vue
                                          22
23
24
25 –
26 \Xi
     main.js
     [o] manifest.json
     [] pages.json
                                                      "text": "直见"

} {

"pagePath": "pages/user/user",

"iconPath": "static/image/user.png",

"selectedIconPath": "static/image/user-active.png",

"text": "用户"
     @ uni.scss
> 🔳 自我练习
                             2、配置pages页
                                          3]
                                                       "navigationBarTextStyle": "black",
                                                      "navigationBarTitleText": "uni-app",
"navigationBarBackgroundColor": "#F8F8F8",
"backgroundColor": "#F8F8F8"
```

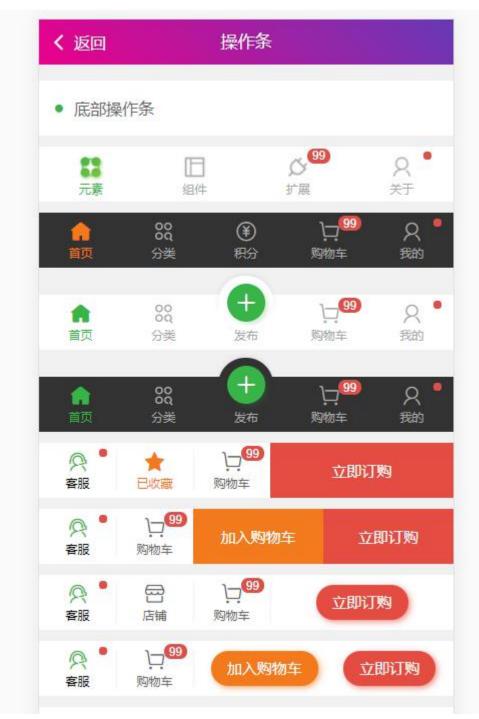
这里我们可以参考 uniapp 官网中的内容, 地址为

https://uniapp.dcloud.io/collocation/pages?id=tabbar

这种方法需要给定图片,而且样式比较局限,因此我们需要知道如何自定义一个 tabbar。

2、使用 colorui 制作自定义 tabbar

我们都知道 colorui 十分强大,它给我们定义了很多样式,比如 tabbar,它给我们定义了很多的 tabbar 样式,如下图所示。

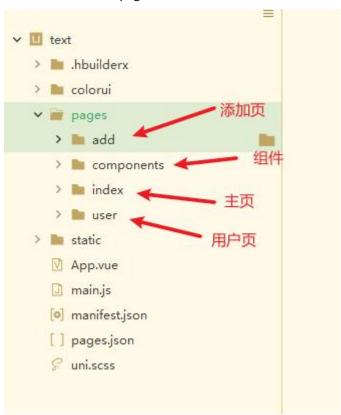


但是这个时候我们就会有很多的疑问,该如何将这些样式放在我们的 tabbar 中呢?这里我们一起来创建第一个 tabbar:

这里我们就使用这个 colorui 中的 tabbar 组件:



第一步: 在我们的 page 页中创建需要的页面:



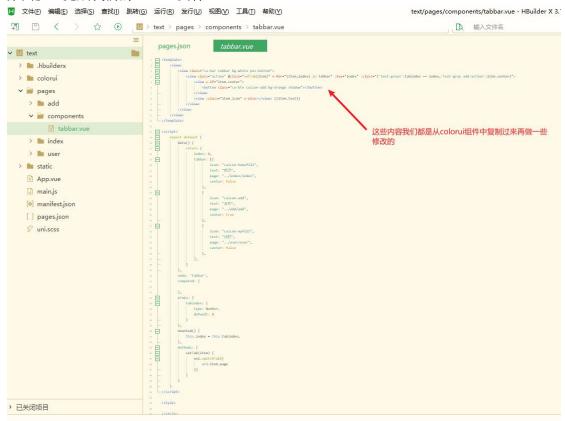
第二步: 配置我们的 page.json 页

```
pages.json
    "path": "pages/index/index",
"style": {
                                                                                                                                              - 优先展示页
  "navigationBarTitleText": "首页"
    "path": "pages/add/add",
"style": {
11 E
                                        "navigationBarTitleText": "添加"
 5 🗏
                                 "path": "pages/user/user",
"style": {
    "navigationBarTitleText": "用户"
 D E
              "tabBar": {
    "color": "87A7E83",
    "selectedColor": "33cc51+",
    "borderStyle": "black",
    "backgroundColor": "#ffffff",
    "list": [{
        """" "pages/index/)
        """ / lng/C
    E
                        "Dackground.cion": "######",
"list": [{
    "pagePath": "pages/index/index",
    "iconPath": "static/img/cart-active.png",
    "selectedIconPath": "static/img/cart-active.png",
    "text": "###"
27 E
                                                                                                                                                                     这里我们虽然设置了tabbar图标,
    "pagePath": "pages/add/add",
"icomPath": "static/img/cart-active.png",
"selectedIcomPath": "static/img/cart-active.png",
"text": "紹介"
                                                                                                                                                                     但是实际上是不需要的,因为后面
                                                                                                                                                                     我们会让它不显示
                       "pagePath": "pages/user/user",
  "iconPath": "static/img/cart-active.png",
  "selectedIconPath": "static/img/cart-active.png",
  "rave". "SEL"
    43 --
45 --
45 --
46 47
48
49
58 --
                       lobalStyle": {
    "navigationBarTextStyle": "black",
    "navigationBarTitleText": "uni-app",
    "navigationBarBackgroundColor": "#F8F8F8",
    "backgroundColor": "#F8F8F8"
```

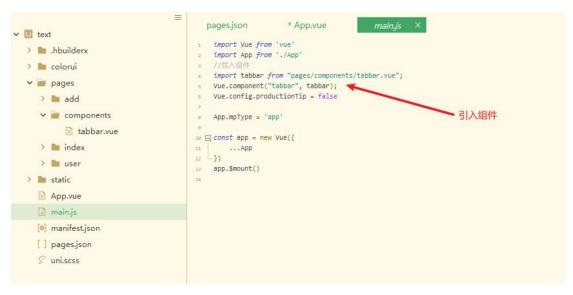
第三步:引入我们的 colorui,配置每个页面的内容,其中这里的 2 和 3 的内容是在我们写好 tabbar 并引用之后使用的,即在第五步之后。

```
a / text / App.vue
                                                                           LY WINXTHO
    pages.json
   1 ⊟ <script>
   3 🗏
          onLaunch: function() {
             console.log('App Launch')
                //隐藏tabbar
                                               ____ 3、这里是让我们原来的tabbar组件隐藏
   5
          uni.hideTabBar()
},
onShow: function() {
               uni.hideTabBar();
   6
  8 =
            console.log('App Show')
  10
            onHide: function() {
  11 🗏
               console.log('App Hide')
  12
  13
  14
  15 -</script>
  17 ⊟ <style>
                                                              _ 1、引入colorui
         /*每个页面公共css */
  18
         @import url("colorui/main.css");
  19
         @import url("colorui/icon.css");
  28
         @import url("colorui/animation.css");
  21
         .pos-bottom{position: fixed !important;bottom: 0 !important;width: 100%;}
                                                                               2、这一步在我们引入
  23
     </style>
                                                                               好tabbar组件后,由
                                                                               于组件会在页面上部
                                                                               展示, 因此我们设置它
                                                                               我们页面的下端
```

第四步: 完成我们的 tabbar 文件:



第五步: 在我们的页面中引用我们的 tabbar 组件



第六步: 在我们的页面使用 tabbar 组件

