

# 2022年夏季《移动软件开发》实验报告

姓名：缪纬韬 学号：22020007160

姓名和学号	缪纬韬，22020007160
本实验属于哪门课程	中国海洋大学24夏《移动软件开发》
实验名称	实验1：第一个微信小程序
Github仓库地址	<a href="https://github.com/spchara/remote-software-develop-lab.git">https://github.com/spchara/remote-software-develop-lab.git</a>

## 一、实验目标

1. 学习使用快速启动模板创建小程序的方法；
2. 学习不使用模板手动创建小程序的方法。

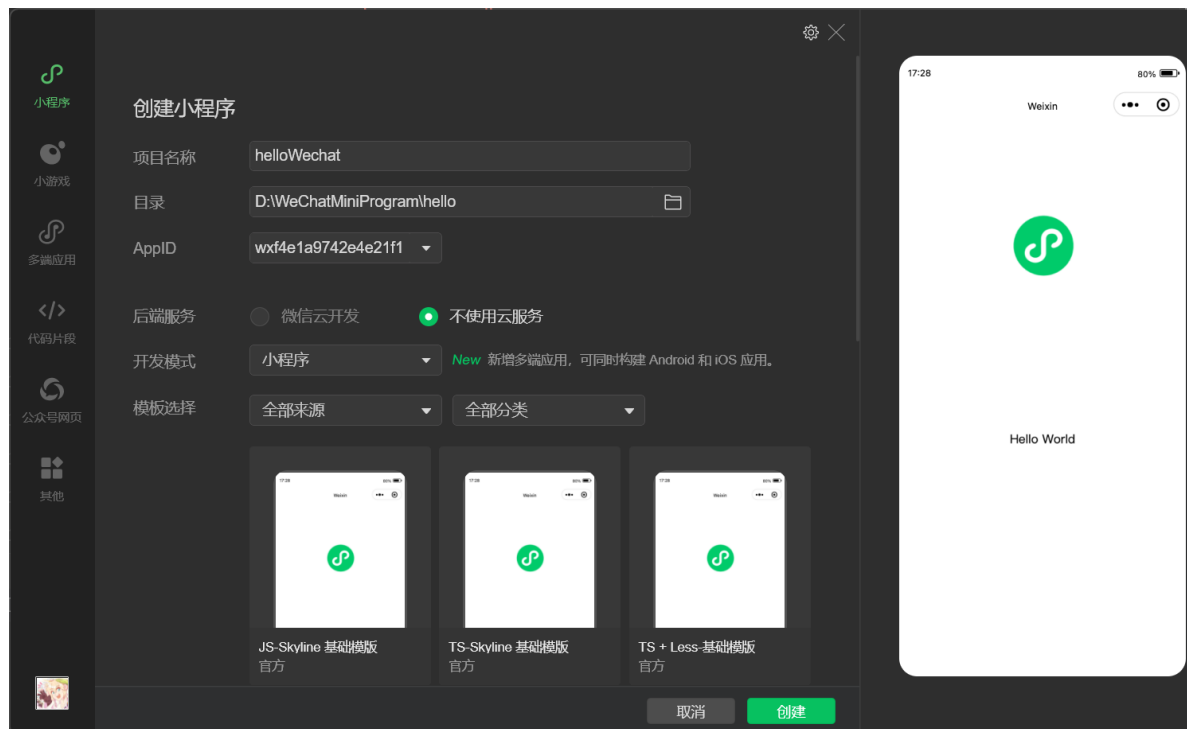
## 二、实验步骤

### 1 模板创建小程序

#### 1.1 创建项目

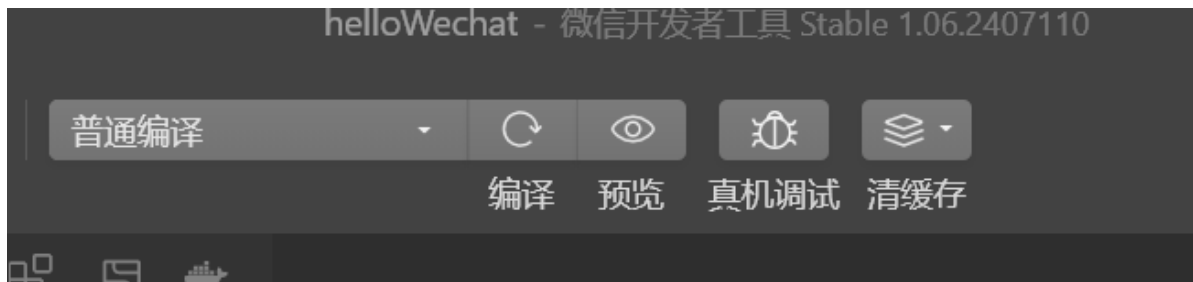
下载安装完成开发者工具后，点击创建小程序

填写一个项目名称，使用一个空文件夹作为项目路径，使用测试号，不使用云服务，最后在模板中选择js-基础模板即可，点击创建




## 1.2 使用

创建完成后，点击最上方的编译，即可在左边的模拟器中看到预览结果



使用效果就是当点击头像时，可以使用自己的头像来替换，点击昵称时，可以选择自动填入微信昵称

16:51

100% 

Weixin



昵称

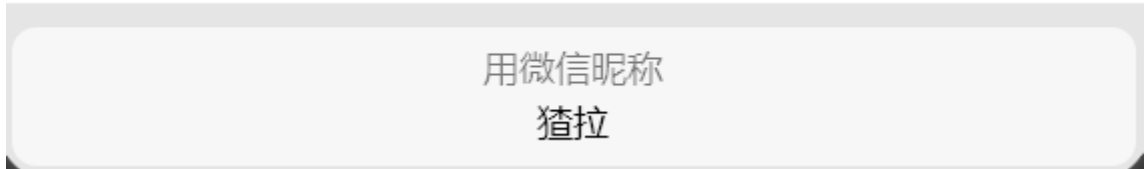
请输入昵称

用微信头像



从相册选择

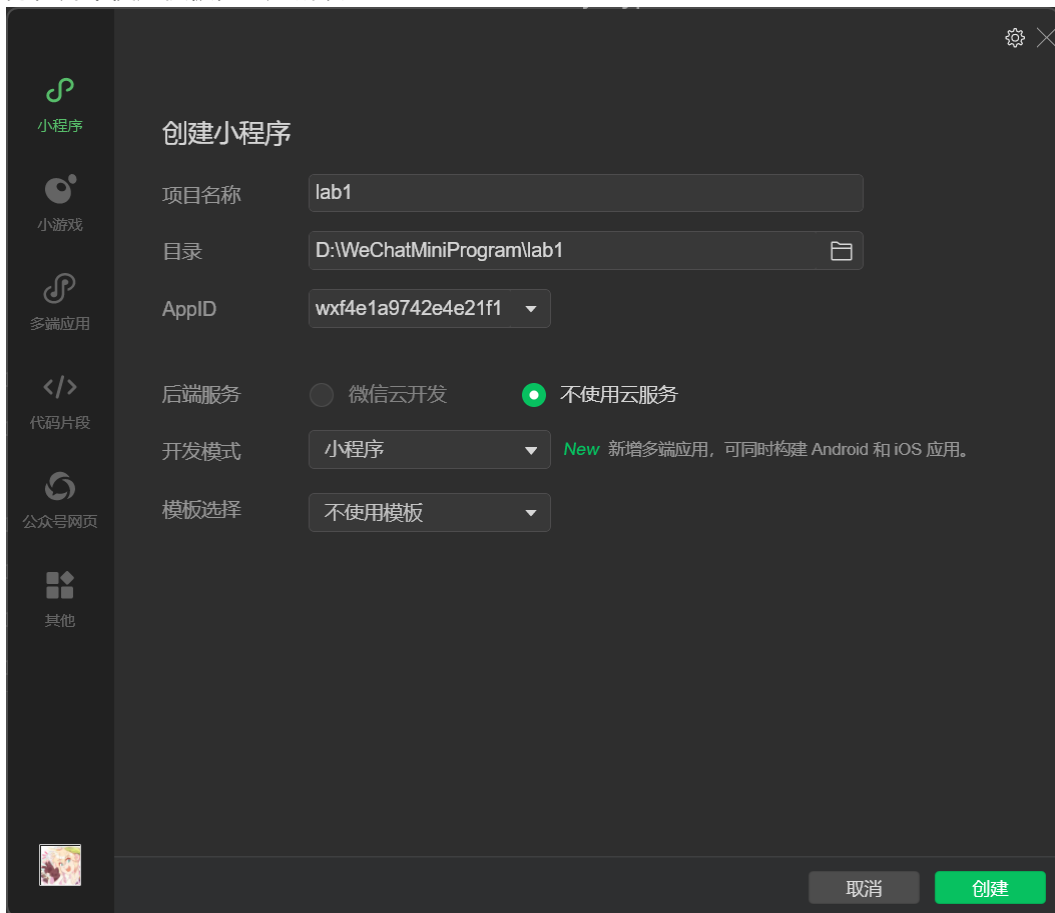
拍照



## 2 手动创建小程序

### 2.1 创建项目

打开微信开发者工具，填写项目名称、项目路径、appid暂时使用测试号，后端服务勾选不适用云服务，并不使用模板，创建新项目



### 2.2 修改导航栏样式

首先按照要求，修改导航栏的背景色，代码如下，用参数 `color` 设置颜色，参数 `background` 设置16进制背景色

```

1  ```.html
2  <navigation-bar title="手动创建第一个程序" back="{{false}}" color="white"
   background="#663399">
3  ```
4
5  ![image-20240819142212834](report_template/assets/image-
   20240819142212834.png)

```

## 2.3 创建视图

接下来完成视图元素的摆放，需要放置一个 `<image>`，`<text>` 和一个 `<button>`

具体代码为：

```

1  <!--index.wxml-->
2  <navigation-bar title="手动创建第一个程序" back="{{false}}" color="white"
   background="#663399">
3  </navigation-bar>
4  <scroll-view class="scrollarea" scroll-y type="list">
5    <view class="container">
6      <view class="userinfo">
7        <image class="avatar" mode="cover" src="{{userInfo.avatarUrl}}">
8      </image>
9      <view class="nickname-wrapper">
10       <text class="nickname-label">昵称</text>
11       <text >{{userInfo.nickName}}</text>
12     </view>
13     <button size="default" type="default" hover-class="button-hover"
   bind:tap="getAvatarAndName">获取头像和昵称</button>
14   </view>
15 </scroll-view>

```

对应的css:

```

1  /**index.wxss*/
2  page {
3    height: 100vh;
4    display: flex;
5    flex-direction: column;
6  }
7  .scrollarea {
8    flex: 1;
9    overflow-y: hidden;
10 }
11
12 .avatar {
13   width: 100px;
14   height: 100px;
15   object-fit: cover;
16   border-radius: 20px;
17   margin: 30rpx;
18 }
19
20 .userinfo {
21   display: flex;

```

```
22     flex-direction: column;
23     align-items: center;
24     color: #aaa;
25     width: 80%;
26 }
27
28 .nickname-wrapper {
29     display: flex;
30     width: 100%;
31     padding: 16px;
32     box-sizing: border-box;
33     border-top: .5px solid rgba(0, 0, 0, 0.1);
34     border-bottom: .5px solid rgba(0, 0, 0, 0.1);
35     color: black;
36 }
37
38 .nickname-label {
39     width: 105px;
40 }
```

15:56

44% 

## 手动创建第一个程序



---

昵称

微信昵称

---

获取头像和昵称

## 2.4 业务逻辑

使用 `bind:tap="getAvatarAndName"` 将获取头像和昵称的js函数绑定到按钮上

在 `index.js` 中我们定义了 `userInfo` 数组对象，包含 `avataoUrl` 和 `nickName` 两个元素，它们一开始用于保存默认的头像（小灰人）以及默认名称（微信昵称）

```
1  data: {
2    userInfo: {
3      avatarUrl: defaultAvatarUrl,
4      nickName: '微信昵称',
5    }
6  },
```

在 `getAvatarAndName` 函数中，我们直接调用wx提供的接口函数 `wx.getUserProfile`，并在成功后将原本的数据元素替换掉

```
1  getAvatarAndName(e) {
2    wx.getUserProfile({
3      desc: '获取头像与昵称',
4      success: (res) => {
5        console.log(res)
6        this.setData({
7          userInfo: res.userInfo,
8        })
9      }
10   })
11 }
```

我们可以通过`console.log`来观察返回的查询结果,其中 `avatarUrl` 就是用户的头像路径了，`nickName` 则是微信昵称

```
1  userInfo:
2    avatarUrl:
3      "https://thirdwx.qlogo.cn/mmopen/vi_32/Q0j4TwGTfTInShcaY9KZ8EG228QUuNmGbmK4f4qAEibs7RFbkmiakZofzV5CXmKCODaUDKShmyNabBRX9UBOCGUA/132"
4    city: ""
5    country: ""
6    gender: 0
7    language: "zh_CN"
8    nickName: "猿拉"
9    province: ""
10   constructor: (...)
11   hasOwnProperty: (...)
12   isPrototypeOf: (...)
13   propertyIsEnumerable: (...)
```



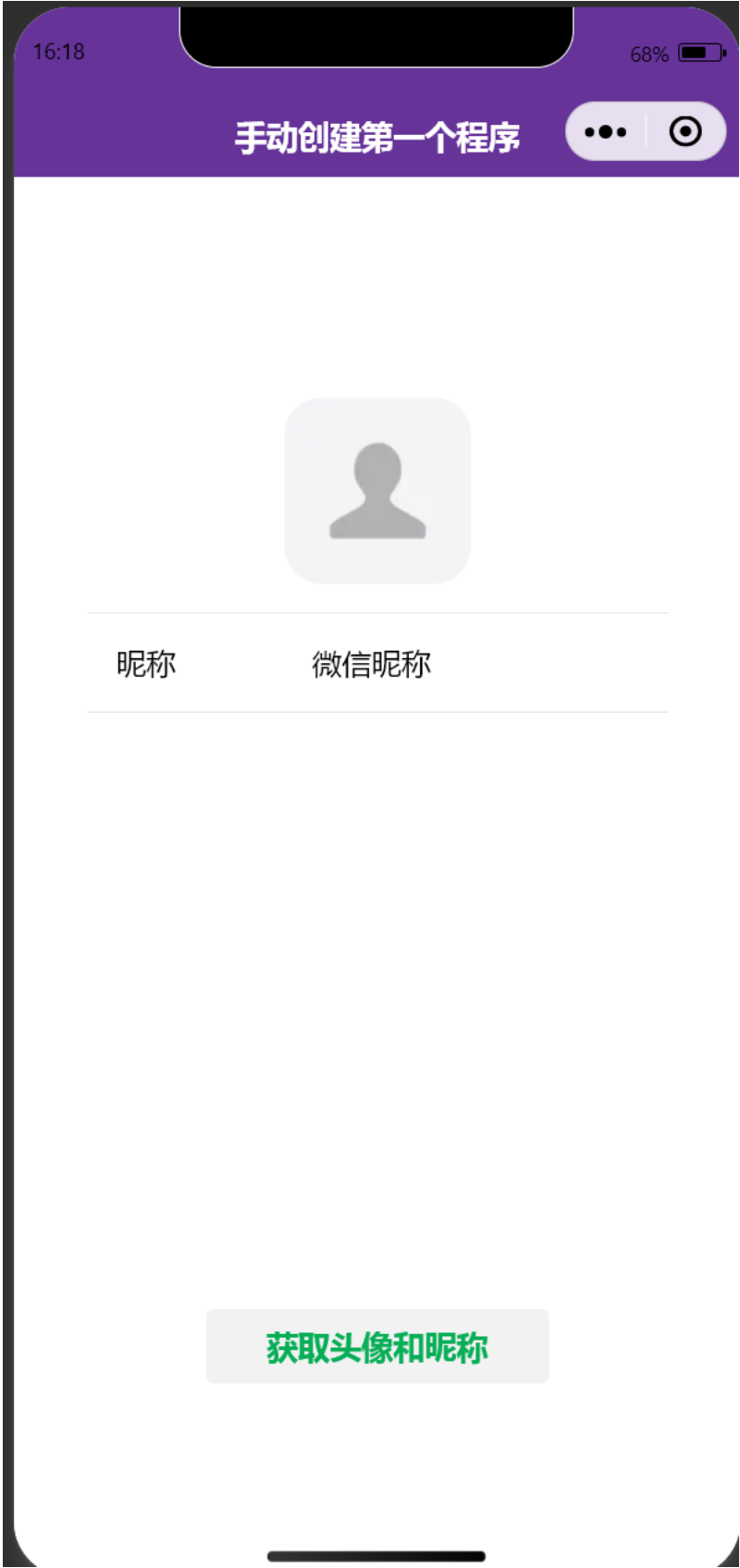
```
13     toLocaleString: (...)  
14     toString: (...)  
15     valueOf: (...)  
16     __defineGetter__: (...)  
17     __defineSetter__: (...)  
18     __lookupGetter__: (...)  
19     __lookupSetter__: (...)
```

至此程序就可以正常运行了

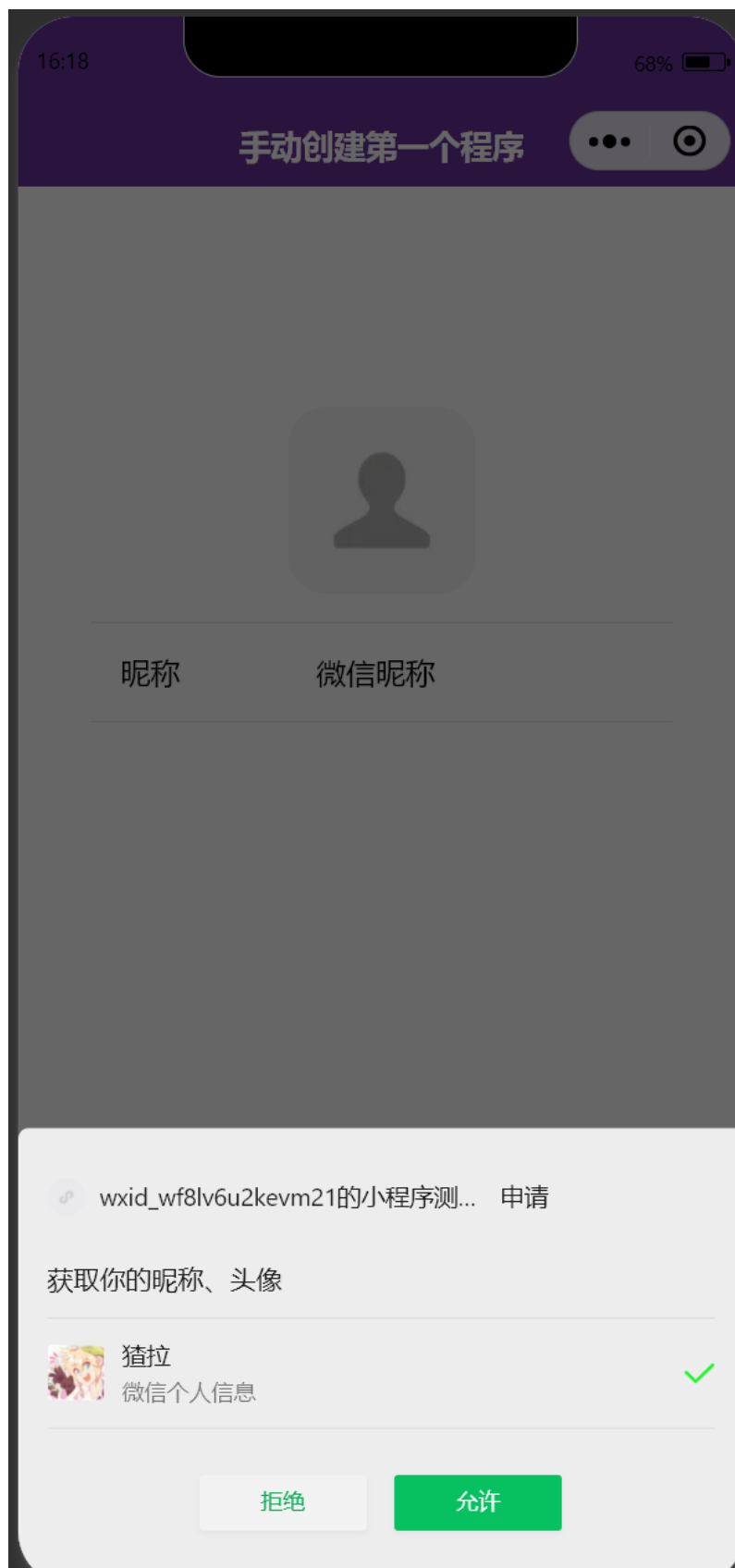
## 三、程序运行结果

---

点击前：



获取请求：



获取成功：



#### 四、程序主要代码

## index.wxml

```
1 <!--index.wxml-->
2 <navigation-bar title="手动创建第一个程序" back="{{false}}" color="white"
  background="#663399">
3 </navigation-bar>
4 <scroll-view class="scrollarea" scroll-y type="list">
5   <view class="container">
6     <view class="userinfo">
7       <image class="avatar" mode="cover" src="{{userInfo.avatarUrl}}">
8     </image>
9     <view class="nickname-wrapper">
10      <text class="nickname-label">昵称</text>
11      <text >{{userInfo.nickName}}</text>
12    </view>
13    <button size="default" type="default" hover-class="button-hover"
  bind:tap="getAvatarAndName">获取头像和昵称</button>
14  </view>
15 </scroll-view>
16
```

## index.wxss

```
1 /**index.wxss*/
2 page {
3   height: 100vh;
4   display: flex;
5   flex-direction: column;
6 }
7 .scrollarea {
8   flex: 1;
9   overflow-y: hidden;
10 }
11
12 .avatar {
13   width: 100px;
14   height: 100px;
15   object-fit: cover;
16   border-radius: 20px;
17   margin: 30rpx;
18 }
19
20 .userinfo {
21   display: flex;
22   flex-direction: column;
23   align-items: center;
24   color: #aaa;
25   width: 80%;
26 }
27
28 .nickname-wrapper {
29   display: flex;
30   width: 100%;
31   padding: 16px;
32   box-sizing: border-box;
33 }
```

```

33 border-top: .5px solid rgba(0, 0, 0, 0.1);
34 border-bottom: .5px solid rgba(0, 0, 0, 0.1);
35 color: black;
36 }
37
38 .nickname-label {
39   width: 105px;
40 }

```

## index.js

```

1  // index.js
2
3  const defaultAvatarUrl =
  'https://mmbiz.qpic.cn/mmbiz/icTdbqWNOWNRna42FI242Lcia07jQodd2FJGIYQfG0LAJGF
  xM4FbnQP6yfMxBgJ0F3YRqJCJ1aPAK2dQagdusBZg/0'
4
5  Page({
6    data: {
7      userInfo: {
8        avatarUrl: defaultAvatarUrl,
9        nickName: '微信昵称',
10     }
11   },
12   onChooseAvatar(e) {
13     const { avatarUrl } = e.detail
14     this.setData({
15       "userInfo.avatarUrl": avatarUrl,
16     })
17   },
18
19   getAvatarAndName(e) {
20     wx.getUserProfile({
21       desc: '获取头像与昵称',
22       success: (res) => {
23         console.log(res)
24         this.setData({
25           userInfo: res.userInfo,
26         })
27       }
28     })
29   },
30 })

```

## 五、问题总结与体会

由于之前接触过一些网页项目，所以初上手微信小程序感觉还是挺熟悉的，不过在实验的过程中遇到了一个问题

一开始我调用 `wx.getUserProfile` 或者 `wx.getUserInfo` 的时候，返回的头像链接固定为小灰人，名称为“微信用户”，如下：

```
1  userInfo:
2    avatarUrl:
3      "https://thirdwx.qlogo.cn/mmopen/vi_32/P0gEwh4mIHO4nibH0KlMECNjjGxQUq24ZEaGT4
4      poC6icRiccVGKSyXwibcPq4BwmiaIGuG1icwxaQX6grC9VemZoJ8rg/132"
5    city: ""
6    country: ""
7    gender: 0
8    is_demote: true
9    language: ""
10   nickName: "微信用户"
11   .....
```

在一番查找之后我发现[小程序用户头像昵称获取规则调整公告 | 微信开放社区 \(qq.com\)](#)发现，微信方面认为在用户刚打开小程序时就要求收集用户的微信昵称头像是不太合理的，因此在后续的标准库更新之后，收回了两个api的功能，**他们将统一返回头像链接为灰色头像，用户名称为“微信用户”**

自 2022 年 10 月 25 日 24 时后（以下统称“生效期”），用户头像昵称获取规则将进行如下调整：

- 1. **自生效期起，小程序 wx.getUserProfile 接口将被收回：**生效期后发布的小程序新版本，通过 wx.getUserProfile 接口获取用户头像将统一返回默认**灰色头像**，昵称将统一返回“微信用户”。生效期前发布的小程序版本不受影响，但如果要进行版本更新则需要进行适配。
- 2. **自生效期起，插件通过 wx.getUserInfo 接口获取用户昵称头像将被收回：**生效期后发布的插件新版本，通过 wx.getUserInfo 接口获取用户头像将统一返回默认**灰色头像**，昵称将统一返回“微信用户”。生效期前发布的插件版本不受影响，但如果要进行版本更新则需要进行适配。通过 wx.login 与 wx.getUserInfo 接口获取 openId、unionId 能力不受影响。
- 3. **「头像昵称填写能力」支持获取用户头像昵称：**如业务需获取用户头像昵称，可以使用「**头像昵称填写能力**」（基础库 2.21.2 版本开始支持，覆盖iOS与安卓微信 8.0.16 以上版本），具体实践可见下方《最佳实践》。
- 4. **小程序 wx.getUserProfile 与插件 wx.getUserInfo 接口兼容基础库 2.27.1 以下版本的头像昵称获取需求：**对于来自低版本的基础库与微信客户端的访问，小程序通过 wx.getUserProfile 接口将正常返回用户头像昵称，插件通过 wx.getUserInfo 接口将正常返回用户头像昵称，开发者可继续使用以上能力做向下兼容。

在更高的版本中，小程序不能直接获取到这些东西了，只能在选择头像的时候，通过在button中添加属性 `open-type=chooseAvatar`，让用户在小程序中主动更改头像时，可以直接选择自己微信的头像。总体来说，就是将主动权交给了用户

那我们该如何进行实验呢？答案就是**调整标准库的版本**，通过把标准库版本跳到2.27.0以下，来使用未被限制的 `wx.getUserProfile` 或者 `wx.getUserInfo`。

	小程序 wx.getUserProfile接口	插件wx.getUserInfo接口	头像昵称填写能力
基础库2.27.1及以上版本	不支持	不支持	支持
基础库2.21.2~2.27.0版本	支持	支持	支持
基础库2.10.4~2.21.0版本	支持	支持	不支持
基础库2.9.5及以下版本	不支持 需进行低版本兼容	支持	不支持



不过如果采用真机调试的话，基础库版本取决于我们的手机，所以真机调试上无法正常运行

当然这种方式只是用于实验，如果是实际开发中还需要跟随着微信的更新来更新自己的业务代码

总体来说，通过今天的实验，我对微信小程序的框架，有了一些具体的认识，接下来继续更深的学习