

# 一、实验准备

课程主页: [课程主页\(gitee.com\)](https://gitee.com)

实验文档: [lab1.pdf\(gitee.com\)](https://gitee.com)

学习视频: [第一个小程序 \(1\) bilibili.com](https://bilibili.com)

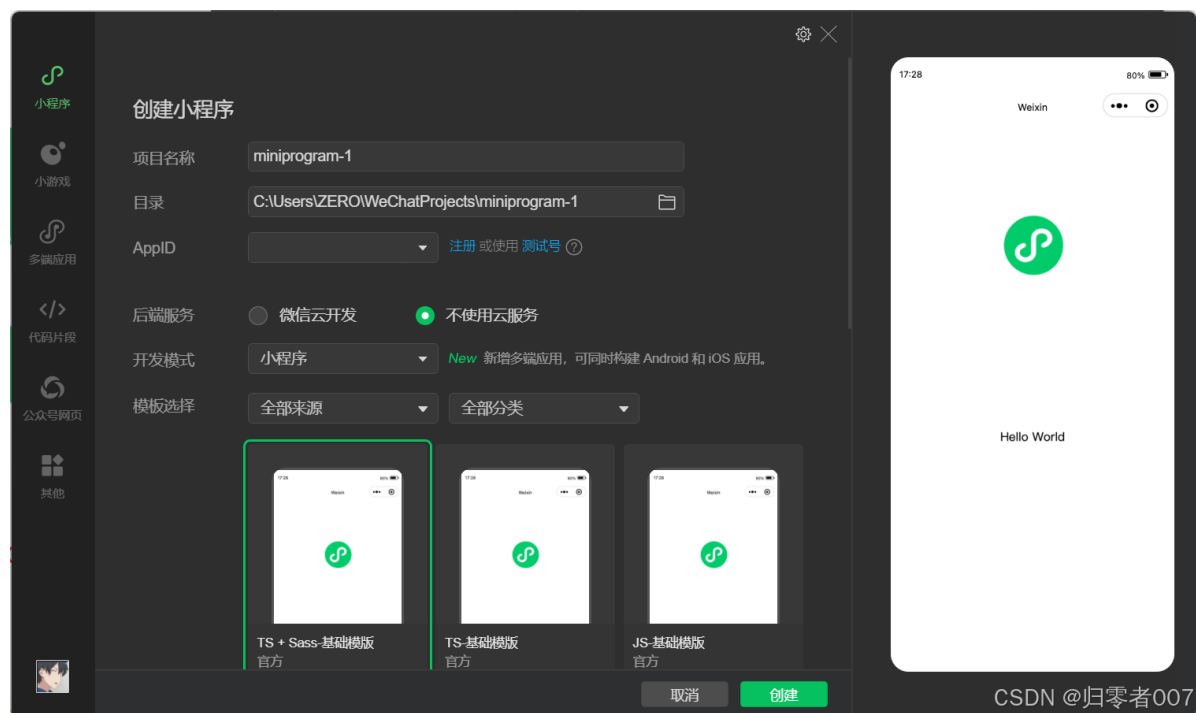
# 二、实验目标

- 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。

# 三、实验方法

## 1、使用模板创建小程序:

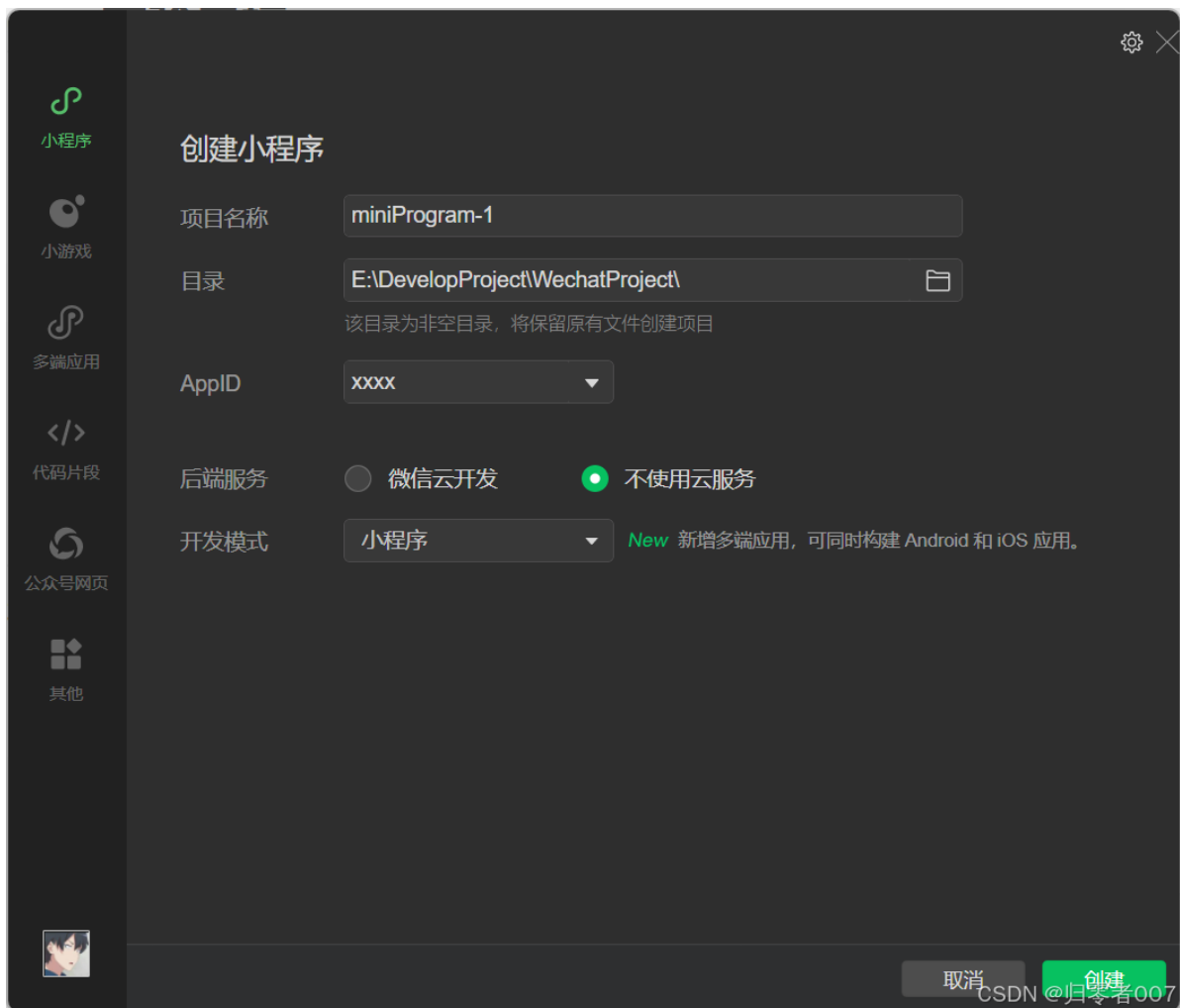
如下图所示, 填写项目名称、目录、AppID以及所使用的后端服务之后, 选择一个所需的模板即可快速创建小程序:



## 2、不使用模板创建小程序:

### 2.1 项目创建

如下图所示, 填写项目名称、目录、AppID以及所使用的后端服务之后, 即可不使用模板创建小程序:



## 2.2 项目配置

### 2.2.1 创建页面文件

项目创建完毕后，在根目录中会生成文件夹pages用于存放页面文件。

具体操作如下：

- (1) 将 app.json 文件内 pages 属性中的“pages/logs/logs”删除，并删除上一行末尾的逗号。
- (2) 按快捷键 Ctrl + S 保存当前修改。

### 2.2.2 删除和修改文件

具体操作如下：

- (1) 删除 utils 文件夹及其内部所有内容。
- (2) 删除 pages 文件夹下的 logs 目录及其内部所有内容。
- (3) 删除 index.wxml 和 index.wxss 中的全部代码。
- (4) 删除 index.js 中的全部代码，并且输入关键词“page”进行自动补全。
- (5) 删除 app.wxss 中的全部代码。
- (6) 删除 app.js 中的全部代码，并且输入关键词“app”进行自动补全。

## 四、实验步骤

列出实验的关键步骤、代码解析、截图。

### 1. 创建项目：

- 使用微信开发者工具创建一个新的小程序项目。
- 配置项目的基本信息，如项目名称、AppID等。

### 2. 编写index.wxml：

```
<!-- pages/index/index.wxml -->
<view class='container'>
  <!-- 显示头像图片 -->
  <image src='{{src}}' class='imageStyle1'></image>
  <!-- 显示昵称 -->
  <text>{{nickName}}</text>
  <!-- 按钮，用于选择头像 -->
  <button open-type="chooseAvatar" bindchooseavatar="getMyInfo">点击获取头像</button>
  <!-- 输入框，用于输入昵称 -->
  <input type="text" placeholder="请输入昵称" bindinput="bindInput" />
</view>
```

### 3. 编写index.wxss：

```
/* pages/index/index.wxss */
.container {
  /* 设置容器高度为视口高度 */
  height: 100vh;
  /* 使用flex布局 */
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: space-around;
}
.imageStyle1 {
  /* 设置图片宽度 */
  width: 400rpx;
  /* 设置图片圆角 */
  border-radius: 50%;
}
.text {
  /* 设置文本字体大小 */
  font-size: 50rpx;
  /* 设置文本颜色 */
  color: red;
}
```

### 4. 编写index.js：

```
// pages/index/index.js
Page({
  data: {
    // 初始化昵称
```

```
    nickName: 'Hello world!',
    // 初始化头像路径
    src: '/images/wechatLogo.jpg',
    newName: ''
  },

  // 获取用户头像信息
  getMyInfo: function(e) {
    console.log("avatarUrl", e.detail.avatarUrl);
    let info = e.detail;
    // 更新昵称和头像
    this.setData({
      nickName: info.nickName,
      src: info.avatarUrl
    });
  },

  // 绑定输入框输入事件
  bindInput: function(e) {
    // 更新昵称
    this.setData({
      nickName: e.detail.value
    });
  },

  /**
   * 生命周期函数--监听页面加载
   */
  onLoad(options) {

  },

  /**
   * 生命周期函数--监听页面初次渲染完成
   */
  onReady() {

  },

  /**
   * 生命周期函数--监听页面显示
   */
  onShow() {

  },

  /**
   * 生命周期函数--监听页面隐藏
   */
  onHide() {

  },

  /**
   * 生命周期函数--监听页面卸载
   */
  onUnload() {
```

```

},

/**
 * 页面相关事件处理函数--监听用户下拉动作
 */
onPullDownRefresh() {

},

/**
 * 页面上拉触底事件的处理函数
 */
onReachBottom() {

},

/**
 * 用户点击右上角分享
 */
onShareAppMessage() {

}
})

```

## 5. 编写App.js:

```

{
  "pages": [
    "pages/index/index",
    "pages/index"
  ],
  "window": {
    "navigationBarTextStyle": "black",
    "navigationBarTitleText": "小程序",
    "navigationBarBackgroundColor": "#09B831"
  },
  "style": "v2",
  "componentFramework": "glass-easel",
  "sitemapLocation": "sitemap.json",
  "lazyCodeLoading": "requiredComponents"
}

```

## 6. 运行项目:

- (1) 在微信开发者工具中运行项目，查看首页的页面效果。
- (2) 测试各个功能是否正常工作，依次检查头像选择功能、昵称输入功能。

# 五、程序运行结果

列出程序的最终运行结果及截图。

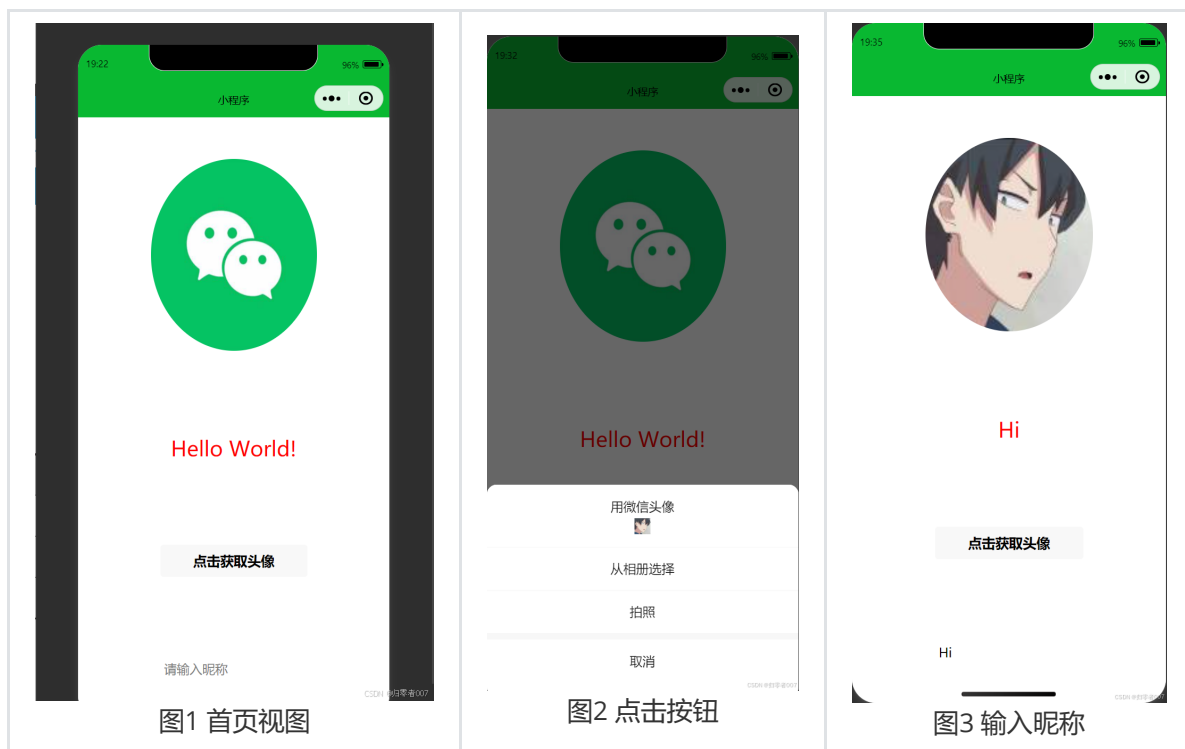
## 1. 页面展示:

- 页面显示一个头像图片、昵称文本、获取头像按钮和输入框。

- 点击按钮可以选择头像，输入框可以输入昵称，昵称会与默认显示“Hello World!”的文本框同时更新。

## 2. 截图：

- 展示页面的截图，显示各个功能的效果。



## 六、问题总结与体会

描述实验过程中所遇到的问题，以及是如何解决的。有哪些收获和体会，对于课程的安排有哪些建议。

### 1. 遇到的问题：

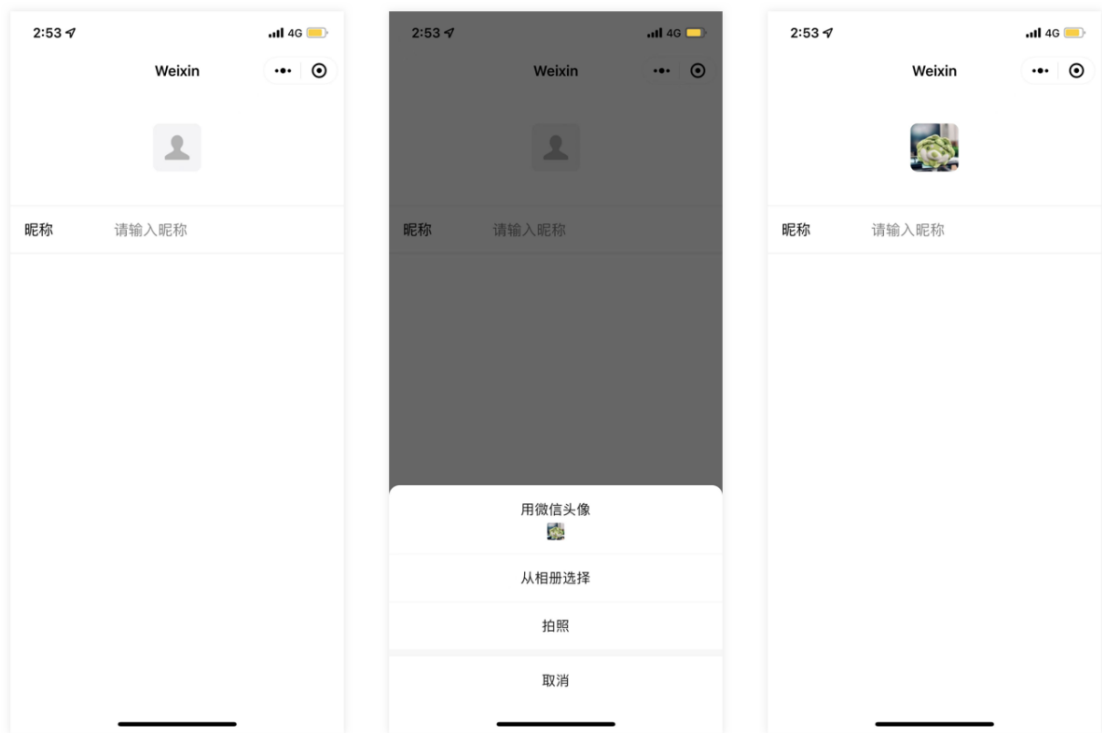
- **问题描述：**使用示例中的“getUserInfo”获取用户信息时失败，导致用户昵称显示为“微信用户”，头像显示为灰色。
- **问题原因：**通过查阅微信官方文档，发现用户头像和昵称的获取规则已经调整。现在需要通过 `getUserProfile` 接口来获取用户信息，并且只能在页面点击事件（例如按钮上的 `bindtap` 回调）中调用。每次请求都会弹出授权窗口，用户同意后会返回 `UserInfo`。

## 使用方法

### 头像选择

需要将 `button` 组件 `open-type` 的值设置为 `chooseAvatar`，当用户选择需要使用的头像之后，可以通过 `bindchooseavatar` 事件回调获取到头像信息的临时路径。

从基础库2.24.4版本起，若用户上传的图片未通过安全监测，不触发 `bindchooseavatar` 事件。

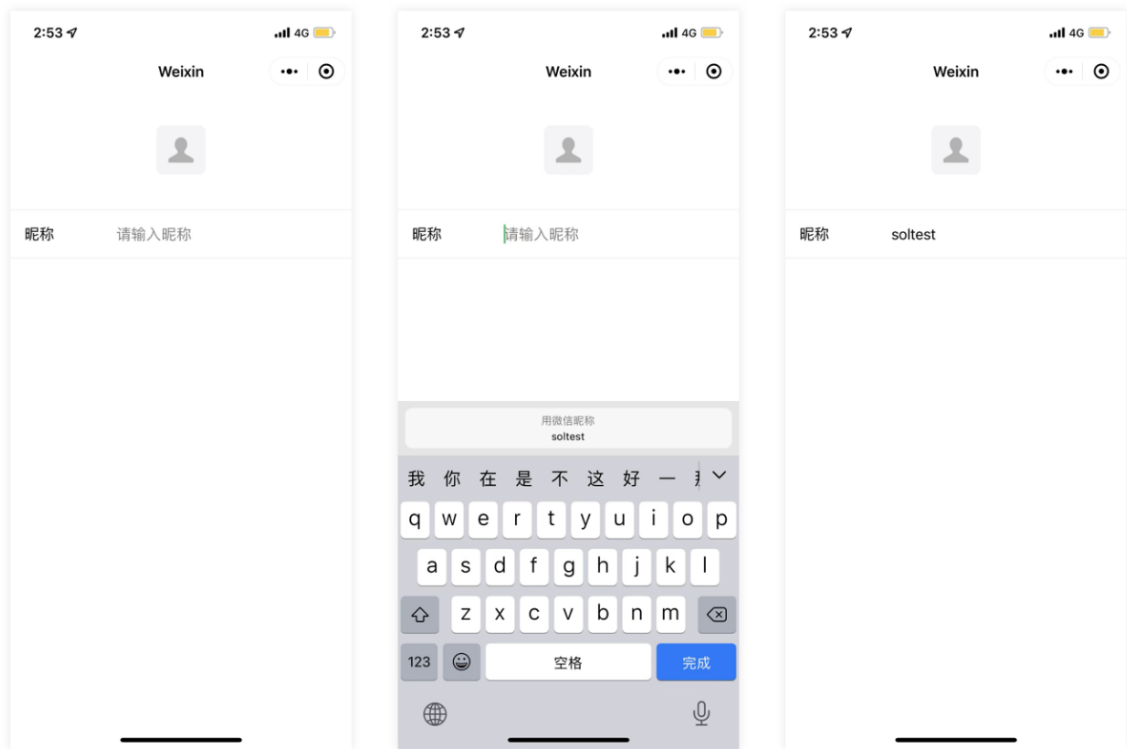


CSDN @归零者007

### 昵称填写

需要将 `input` 组件 `type` 的值设置为 `nickname`，当用户在此input进行输入时，键盘上方会展示微信昵称。

从基础库2.24.4版本起，在 `onBlur` 事件触发时，微信将异步对用户输入的内容进行安全监测，若未通过安全监测，微信将清空用户输入的内容，建议开发者通过 `form` 中 `form-type` 为 `submit` 的 `button` 组件收集用户输入的内容。



CSDN @归零者007

## 2. 解决方法:

- index.wxml文件中使用如下:

```
<!-- 按钮, 用于选择头像 -->
<button open-type="chooseAvatar" bindchooseavatar="getMyInfo">点击获取头像</button>
<!-- 输入框, 用于输入昵称 -->
<input type="text" placeholder="请输入昵称" bindinput="bindInput" />
```

- index.js文件中使用如下:

```
// 获取用户头像信息
getMyInfo: function(e) {
  console.log("avatarUrl", e.detail.avatarUrl);
  let info = e.detail;
  // 更新昵称和头像
  this.setData({
    nickName: info.nickName,
    src: info.avatarUrl
  });
},

// 绑定输入框输入事件
bindInput: function(e) {
  // 更新昵称
  this.setData({
    nickName: e.detail.value
  });
},
```

## 3. 收获与体会:

这次实验作为小程序开发的入门, 通过编写一个获取微信头像和昵称的小demo, 展示了如何从零开始设计一个微信小程序, 从而熟悉了小程序的基本开发框架和常用组件的使用。