第3章

豆豆云助教“我的”页面模块开发


从此我不再仰脸看青天，不再低头看白水，只谨慎着我双双的脚步，我要一步一步踏在泥土上，打上深深的脚印。

——朱自清

不积跬步，无以至千里；不积小流，无以成江海。

曾经有一个少年，向一位大师求教如何才能做到跋山涉水不费吹灰之力，大师没有直接传授给他技艺，只叫他每天为自己养猪，并且有一个要求，必须抱着猪越过一座座山坡，翻过一条条河沟，晚上再抱回家，中途不能放下猪。日子一天天过去，少年心中不满，心想大师怎么每天只让自己为他养猪，但碍于情面也就没有吱声。两年后的某一天，大师突然对他说：“今天你不必抱着猪，自己上山去吧。”少年满心疑惑，但也照做了。意外的是，一路上他只觉得身轻如燕，脚步飞快，不一会儿便到了山顶。

少年恍然大悟，在过去的两年里，小猪仔一天天长大，从几斤长到了两百多斤，而自己每天抱着猪上山已经练就了一身的爬山好本领。

学习也是如此，点滴积累会让我们在不知不觉中变成高手，小程序开发的学习正式开始了，望同学们持之以恒，同时也要对自己充满信心。本章主要通过“我的”页面模块开发正式开始豆豆云助教的开发。完成“我的”页面模块开发之前，需要先完成授权登录页面和注册页面，拥有授权信息与注册信息后，才能在“我的”页面将个人信息显示出来。

3.1 授权登录页面

本节主要分为两部分，首先讲解授权登录页面涉及的知识点，然后在理解的情况下完成授权登录页面的开发。

3.1.1 授权页面知识点讲解

1. 小程序登录

小程序可以通过微信官方提供的登录功能方便地获取微信提供的用户身份标识，快速建立小程序内的用户体系。如图 3-1 所示，小程序通过 wx.login()获取 code(登录凭证)，然后通过 wx.request()发送 code 至开发者服务器，开发者服务器将登录凭证 AppID、appsecret 与 code 用于校验微信接口，微信接口服务向开发者服务器返回用户唯一标识 openid 和会话密钥 session_key。开发者服务器实现自定义登录状态与 openid、session_key 的关联，并向小程序返回自定义状态。小程序将自定义登录状态存入 storage，并用于后续 wx.request 发起业务请求。

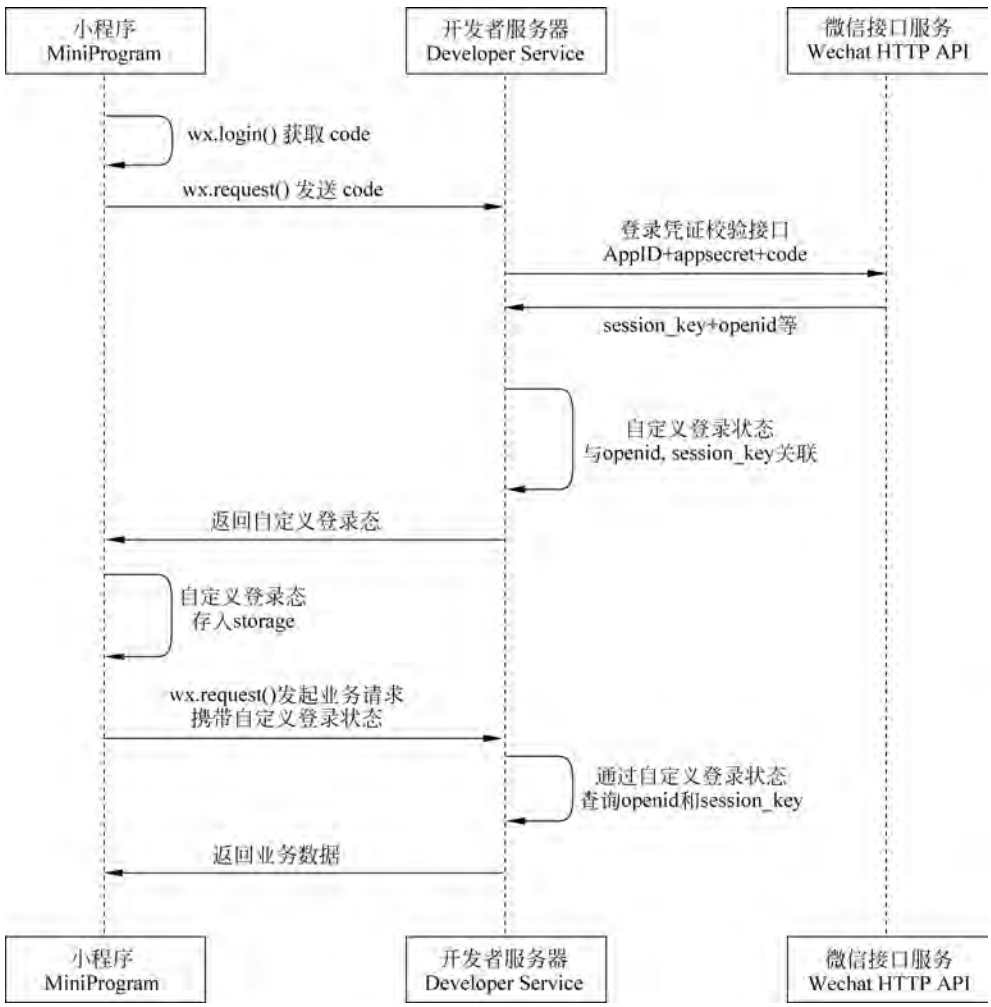

图 3-1 小程序登录流程时序

对于某个微信小程序，每个用户访问该小程序都会产生一个唯一的 openid，这个 openid 为用户访问该小程序的标识符，即每个用户的 openid 都是不一样的。因此，可以把 openid 作为用户唯一标识符（类似身份证号），并存于数据库中用于后续操作。

开发者服务器与微信接口服务之间的交互是由后台实现的，本节主要以小程序前端与开发者服务器之间的交互为主，后台部分会在第 9 章中详细介绍。

2. wx.login()

调用 wx.login()接口获取 code，通过 code 进而换取用户登录状态信息，其中 wx.login()接口属性如表 3-1 所示。

表 3-1　wx.login()接口属性

属　性	类　型	必　填	说　明
timeout	number	否	超时时间，单位为 ms
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数（无论调用成功还是失败都会执行）

由于 app.js 会先于其他页面执行，所以比较适合处理一些注册函数，因此将 wx.login()方法写在 app.js 文件中。

3. wx.request()

wx.request()主要用于发送 HTTPS 网络请求，其属性详见表 3-2。

表 3-2　wx.request 属性

属　性	类　型	默认值	必填	说　明
url	string		是	开发者服务器接口地址
data	string/object/ArrayBuffer		否	请求参数
header	object		否	设置请求的 header，header 中不能设置 Referer。content-type 默认为 application/json
method	string	GET	否	HTTP 请求方法
dataType	string	json	否	返回的数据格式
responseType	string	text	否	响应的数据类型
success	function		否	接口调用成功的回调函数
fail	function		否	接口调用失败的回调函数
complete	function		否	接口调用结束的回调函数（无论调用成功还是失败都会执行）

这里以小程序登录中小程序向开发者服务器发送 wx.request 请求为例，调用微信官方的 wx.login()接口会返回一串 jscode，服务器使用 jscode、AppID、appsecret 三个参数向微信请求得到 openid，这一步后台已经封装完成，并提供一个开放接口：

https://zjgsujiaoxue.applinzi.com/index.php/Api/Weixin/code_to_openidv2

具体代码如下：

```
//登录
wx.login({
      success: res => {
        //发送 res.code 到后台换取 openid, session_key, unionid
        wx.request({
          url:'https://zjgsujiaoxue.applinzi.com/index.php/Api/Weixin/code_to_openidv2',
          data: {
            'code': res.code,
            'from': 'wxbf9778a9934310a1'
          },
          success:function (res) {
            console.log(res.data)
            //将 sessionid 保存到本地 storage
            wx.setStorageSync('jiaoxue_OPENID', res.data.openid)
          },
          fail:function (res) {
            console.log('res' + res)
          }
        })
      }
    })
```

上述代码中，通过 wx.login()方法，成功返回 res，其中 res.code 为微信官方返回的 code，通过 wx.request()发起请求，请求参数为 code 与 appid，当请求成功时，后台会返回一个数组，数组中包含的值是由后台代码决定的，其中就包含了 openid，这里可以使用 console.log(res.data)来看一下返回的数组中所包含的值，如图 3-2 所示。

图 3-2　wx.request()请求的返回值

4. 数据缓存

每个微信小程序都可以有自己的本地缓存，通过数据缓存 API 可以对本地缓存进行设置、获取和清理。同一个微信用户，同一个小程序 storage 上限为 10MB。localStorage 以用户维度隔离，同一台设备上，A 用户无法读取到 B 用户的数据。

注意：如果用户存储空间不足，微信会清空最近且最久未使用的小程序的本地缓存。因此不建议将关键信息全部存在 localStorage，以防储存空间不足或用户换设备的情况。

数据缓存 API 主要有五类，包括数据的存储、获取、移除、清空以及获取存储信息，每类均包含同步与异步两种，具体详见表 3-3。

表 3-3　数据缓存 API 函数类型

函　数　名	说　　明
wx. setStorage(Object object)	数据的存储(异步)
wx. setStorageSync(string key, any data)	数据的存储(同步)
wx. getStorage(Object object)	数据的获取(异步)
wx. getStorageSync(string key)	数据的获取(同步)
wx. getStorageInfo(Object object)	存储信息的获取(异步)
wx. getStorageInfoSync()	存储信息的获取(同步)
wx. removeStorage(Object object)	数据的移除(异步)
wx. removeStorageSync(string key)	数据的移除(同步)
wx. clearStorage(Object object)	数据的清空(异步)
wx. clearStorageSync()	数据的清空(同步)

其中，Sync 为英文单词 synchronization 的前四个字母，表示同步，因此 API 函数中带有 Sync 后缀的函数为同步函数。同步函数与异步函数之间的区别是，异步函数不会阻塞当前任务，同步函数缓存直到同步方法处理完才能继续往下执行。另外异步函数中含有成功回调函数，可用于数据处理成功后的操作。

这里以 wx. login() 中使用的 wx. setStroageSync() 为例，将 wx. request() 返回的 openid 存储于本地，方便 openid 的获取。使用 wx. setStorageSync()的代码示例如下：

```
wx.setStorageSync('jiaoxue_OPENID', res.data.openid)
```

编译后，可以在调试器的 Storage 面板中看到 openid 已存入本地，Key 的值为 jiaoxue_OPENID，Value 的值为用户的 openid，如图 3-3 所示。

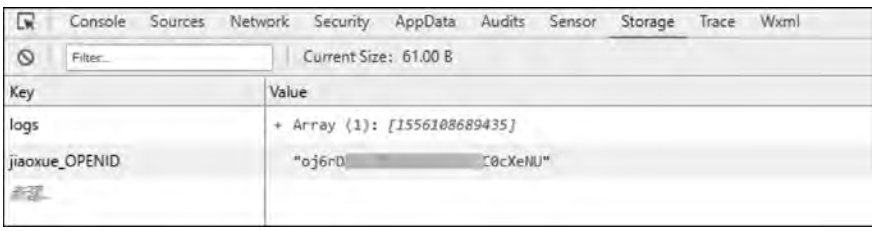

图 3-3　Storage 面板中的本地缓存

如果使用 wx. setStorage()进行数据存储，可以对数据存储成功后再进行操作，代码较 wx. setStorageSync()有变化，具体代码如下：

```
wx.setStorage({
    key:'jiaoxue_OPENID',
    data:res.data.openid,
```

```
        success:function(){
            console.log('存储成功')
        }
    })
```

编译后，同样将 openid 存储于本地缓存，并执行成功回调函数，Console 面板打印出“存储成功”，如图 3-4 所示。

图 3-4　Console 面板中的“存储成功”

需要使用本地缓存中的 openid 时，可以用 wx.getStorageSync('jiaoxue_OPENID')从本地获取 openid，并赋值给相应的变量。当然 wx.getStorage()也可以，这里不赘述。

5. wx.showModal()

小程序使用 wx.showModal(Object object)显示模态对话框，其中 object 参数说明如表 3-4 所示。

表 3-4　wx.showModal()中 Object 参数说明

属　性	类　型	默认值	必填	说　明
title	string		是	提示的标题
content	string		是	提示的内容
showCancel	boolean	true	否	是否显示取消按钮
cancelText	string	'取消'	否	取消按钮的文字，最多 4 个字符
cancelColor	string	#000000	否	取消按钮的文字颜色，必须是十六进制格式的颜色字符串
confirmText	string	'确定'	否	确认按钮的文字，最多 4 个字符
confirmColor	string	#576B95	否	确认按钮的文字颜色，必须是十六进制格式的颜色字符串
success	function		否	接口调用成功的回调函数
fail	function		否	接口调用失败的回调函数
complete	function		否	接口调用结束的回调函数(无论调用成功还是失败都会执行)

其中 success()回调函数的返回参数详见表 3-5。

表 3-5　success()回调函数的返回参数

属　性	类　型	说　明	最低版本
confirm	boolean	为 true 时，表示用户单击了“确定”按钮	
cancel	boolean	为 true 时，表示用户单击了“取消”按钮(用于 Android 系统区分单击“蒙层”关闭还是单击“取消”按钮关闭)	1.0.0

在进入豆豆云助教时，如果用户没有注册过，会弹出模态对话框提示用户前往注册，具体代码如下：

```
if (!res.data.is_register) {
    wx.showModal({
        title:'提示',
        content:'请先注册',
        showCancel:false,
        confirmText:"确定",
        success:function(res) {
            wx.navigateTo({
            url:'/pages/register/userlogin',
            })
        }
    })
}
```

编译后，弹出模态对话框，提示用户前往注册，如图 3-5 和图 3-6 所示。

图 3-5　模态对话框提示用户注册（不含“取消”按钮）

图 3-6　模态对话框提示用户注册（含“取消”按钮）

另外尝试在该 wx.showModel()的基础上，进行简单的修改，首先将 showCancel 属性删除，这样模态对话框会默认 showCancel 的值为 true。然后添加一个成功回调函数 success()，通过 console.log()查看 success()的返回值具体有哪些，具体代码如下。

```
wx.showModal({
    title:'提示',
    content:'请先注册',
    confirmText:"确定",
    success:function(res) {
        console.log(res)
        if(res.confirm){
          console.log('"确定"按钮被单击')
          wx.navigateTo({
            url:'/pages/register/userlogin',
        })
        }else if(res.cancel){
          console.log('"取消"按钮被单击')
        }
    }
})
```

编译后,效果如图 3-6 所示。在 Console 面板中可以看到打印出来的 success()函数的返回值,如图 3-7 所示。

图 3-7　success()函数的返回值

3.1.2　授权登录页面实现

1. 新建小程序项目

首先新建一个小程序项目,具体操作与 1.1.3 节中 Hello World 小程序的新建一样,新建项目时,建议开发者自定义项目名称,并且在存放小程序项目的目录下新建一个空的文件夹,项目目录选择该文件夹,这样方便以后寻找项目所在目录。项目名称可自定义,本书将项目名称命名为 doudouyun,与项目相关,具体如图 3-8 所示。

2. 新建 userlogin 页面

完成项目新建后,需要新建一个授权登录页面,首先右击 pages 目录,在弹出的快捷菜单中选择“新建目录”命令,新建目录并命名为 register。然后右击 register 目录,在弹出的

快捷菜单中选择“新建 Page”命令，新建 Page 并命名为 userlogin，如图 3-9 和图 3-10 所示。

图 3-8　新建 doudouyun(豆豆云)项目

图 3-9　选择“新建目录”命令

图 3-10　选择“新建 Page”命令

选择“新建 Page”命令而不选择一个一个文件新建，原因是选择“新建 Page”命令时，app.json 的 pages 属性中会自动添加新建的页面，开发者不需要再手动添加页面路径了。

3. userlogin 页面开发

userlogin 页面的功能主要是授权，与 Hello World 小程序中 index 页面的功能相似，因此只要在 Hello World 小程序的基础上进行简单修改即可。

首先是 wxml 文件。userlogin 页面结构主要由 view、text 与 button 三种标签组成，并使用 class 属性定义对应标签的样式，页面中主要有一个“单击授权登录”按钮，具体代码如下：

```
<!-- userlogin.wxml -->
<view class = "container">
<view class = "usermotto">
<text class = "user - motto">微信授权</text >
</view >
<view class = "userinfo">
<button wx:if = "{{!hasUserInfo && canIUse}}" open - type = "getUserInfo" bindgetuserinfo =
"getUserInfo">单击授权登录</button >
</view >
</view >
```

然后是 wxss 文件。相比于 Hello World 小程序中的 index.wxss 文件，少了两种样式类型，主要保留了 userinfo 与 usermotto，具体代码如下：

```
/ ** userlogin.wxss ** /
.userinfo {
display: flex;
flex - direction: column;
align - items: center;
}
.usermotto {
margin - top: 150px;
text - align: center;
}
```

为了获得更好的用户体验，一些细节也要注意一下，比如当用户进入授权登录页面时，页面导航栏的标题文字也相应变为“授权页面”，主要就是在 json 文件中加上一行代码，具体代码如下：

```
{
"navigationBarTitleText": "授权页面"
}
```

最后就是 userlogin.js 中的相关逻辑代码。userlogin 页面的逻辑与 Hello World 小程序中 index 页面的逻辑基本一样，只是简单调整了一下，原有的事件处理函数 bindViewTap()在授权页面不需要了，直接删除即可。在 onLoad()函数最后加上一个判断语句，判断当 hasUserInfo!=false 时，跳转至 register 页面，即注册页面，具体代码如下：

```
if (this.data.hasUserInfo) {
    wx.navigateTo({
      url:'./register',
    })
}
```

另外 getUserInfo()函数中也相应加上一个页面跳转函数 wx.navigateTo()，实现当触

发事件处理函数 getUserInfo()时，跳转至 register 页面，具体代码如下：

```
getUserInfo:function (e) {
    wx.navigateTo({
      url:'./register',
    })
    app.globalData.userInfo = e.detail.userInfo
    this.setData({
      userInfo: e.detail.userInfo,
      hasUserInfo:true
    })
}
```

最后授权登录页面的效果如图 3-11 所示。

图 3-11　授权登录页面效果

如果之前已经授权过了，看不到想要的授权页面，可以单击工具栏中间区域的“清缓存”按钮清除授权记录。

4. app.js

除了完成 userlogin 页面的开发，还需要对 app.js 文件进行修改。首先是 wx.login()方法需要完善，这样才能实现小程序的登录功能，最终代码如下：

```
wx.login({
```

```
success: res => {
  //发送 res.code 到后台换取 openid, session_key, unionid
  wx.request({
    url:'https://zjgsujiaoxue.applinzi.com/index.php/Api/Weixin/code_to_openidv2',
    data: {
      'code': res.code,
      'from': 'wx5ee2da791099a208'
    },
    success:function (res) {
      console.log(res.data)
      //将 sessionid 保存到本地 storage
      wx.setStorageSync('jiaoxue_OPENID', res.data.openid)
      if (!res.data.is_register) {
        wx.showModal({
          title:'提示',
          content:'请先注册',
          showCancel:false,
          confirmText:"确定",
          success:function (res) {
            wx.navigateTo({
              url:'/pages/register/userlogin',
            })
          }
        })
      }
    },
    fail:function (res) {
      console.log('res' + res)
    }
  })
}
})
```

注意，wx.request()的 data 数组中，from 对应的是开发者的 appid，因此 appid 的值需要改成开发者自己的 appid。

编译后，发现 Console 面板会提示错误，如图 3-12 所示。

图 3-12　提示 request 中 url 不在合法域名列表

解决方法：单击工具栏右侧区域的“详情”按钮，勾选“不校验合法域名”复选框即可，如图 3-13 所示。

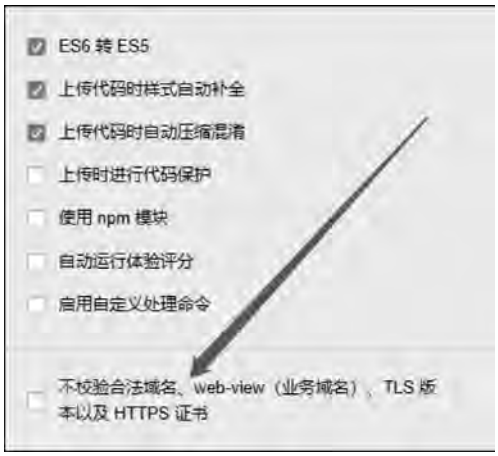

图 3-13　勾选“不校验合法域名”复选框

勾选“不校验合法域名”复选框后，重新编译一次，发现 Console 面板提示“该 appid 未注册”，如图 3-14 所示。

图 3-14　提示“该 appid 未注册”

这是为了让所有开发者在学习豆豆云前端开发时，使用提供给所有开发者的云后台。豆豆云为开发者专门提供了一个接口，前往注册一下，即可使用提供的云后台。因此要使 wx.login 方法的 wx.request()中的 url 实现访问后台，需要前往 https://zjgsujiaoxue.applinzi.com/index.php/Page/Index/register 进行注册。调用该接口需要两个参数，即开发者的 appid 与 appsecret，如图 3-15 所示。

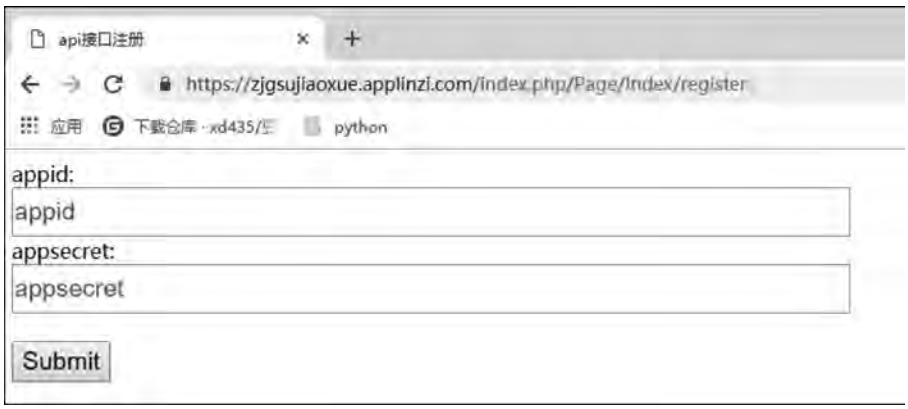

图 3-15　API 接口注册

填写 appid 与 appsecret 后，单击 Submit 按钮即可完成 API 接口注册。API 接口注册完成后，重新编译代码即可看到 Console 面板中 wx.request()的返回值，主要包括 is_login、is_register 和 openid，如图 3-16 所示。

图 3-16　wx.request()返回值

到这里，用户第一次进入授权登录页面的跳转逻辑已经完成。

3.2 注册页面

如果要在 userlogin 的逻辑中跳转到注册页面，就需要新建一个 register 页面。本节主要先对注册页面中的一些知识点进行讲解，然后再具体介绍如何完成注册页面的开发。

3.2.1 注册页面知识点讲解

注册页面主要新增了三个知识点，分别是微信官方 UI 库 WeUI、bindchange 事件和 openAlert()函数。

1. 微信官方 UI 库 WeUI

WeUI 是一套同微信原生视觉体验一致的基础样式库，由微信官方设计团队为微信内网页和微信小程序量身设计，令用户的使用感知更加统一。它包含 button、cell、dialog、progress、toast、article、actionsheet、icon 等各种元素。WeUI 基础样式库下载地址为 https://github.com/Tencent/weui-wxss。开发者可以将样式库下载并使用微信 Web 开发者工具打开 dist 目录(请注意，是 dist 目录，不是整个项目)，导入 dist 目录后，可以预览样式库，如图 3-17 所示。

开发者可以在样式库里选择自己所需要的样式，然后直接将需要的样式对应的 wxml 代码复制、粘贴至自己的项目中，然后将 WeUI 中 style 文件复制至自己的项目目录中，如将图 3-18 目录下 style 文件夹复制至图 3-19 目录下。

将 style 文件夹复制至自己开发的项目后，还需要在 app.wxss 文件中使用@import 导入 WeUI 的样式，如图 3-20 所示。到这里，即可正常使用 WeUI 库中微信的官方样式。

图 3-17　预览 WeUI 样式库

图 3-18　dist 目录下的 style 文件夹

图 3-19　doudouyun 项目下的 style 文件夹

图 3-20　导入 WeUI 样式

2. bindchange 事件

bindchange 事件与 bindtap 事件不同，它主要是当输入框中的内容发生改变时，触发对应的事件处理函数，并且输入框中的值可以通过 event.detail.value 来获取。举个简单的例子，代码如下。

wxml 文件代码：

```
<view class="weui-cells weui-cells_after-title">
  <view class="weui-cell weui-cell_input">
    <view class="weui-cell__hd">
      <view class="weui-label">qq</view>
    </view>
    <view class="weui-cell__bd">
    <input class="weui-input" placeholder="请输入 qq" bindchange="changevalue"/>
    </view>
  </view>
</view>
```

js 文件代码：

```
Page({
  data: {
    qq:0
  },
  changevalue:function(event){
    console.log(event)
    this.setData({
      qq: event.detail.value
    })
  },
})
```

页面效果如图 3-21 所示。

图 3-21　bindchange 使用样例

当在输入框中输入内容后，单击其他空白处，可以打印出 changevalue()函数的返回值，会发现输入的内容被存放在 detail 的 value 中，如图 3-22 所示。

图 3-22　bindchange 事件触发后 value 的值

3. openAlert ()函数

openAlert()函数是在 js 文件中自定义的一个函数，在定义函数后，可以在其他函数中使用 this. openAlert()调用 openAlert()函数。

3.2.2　注册页面实现

注册页面实现主要分为两部分：一部分是注册页面的页面布局；另一部分则是注册页面的功能实现。

1. 注册页面的页面布局

与新建 userlogin 页面一样，在 register 目录下，右击 register，在弹出的快捷菜单中选择“新建 Page”命令新建 Page 并命名为 register。建完 register 页面后，接下来就是往页面里写东西了。

首先看 register 页面最后的界面需要做成什么样，如图 3-23 所示。

然后在 WeUI 基础样式库中找到对应的样式，其中姓名、手机号、学校、学号和入学年

份是一个输入框，对应的是 WeUI 中表单下 Input 里面的一种样式，如图 3-24 所示。单击模拟器下方的“打开”按钮，即可在编辑器的目录结构区找到该页面对应的目录，打开 input.wxml 文件，找到该样式对应的代码，如图 3-25 所示。将其复制至 doudouyun 项目的 register.wxml 中，其中这段代码最后还少了一个</view>，作为最开始<view>的结束标签。

图 3-23　注册页面效果

图 3-24　WeUI 样式库中对应的 Input 样式

以姓名的 input 为例，其他项都与姓名的操作一致，register.wxml 代码如下：

```
<view class="weui-cells weui-cells_after-title">
  <view class="weui-cell weui-cell_input">
    <view class="weui-cell__hd">
      <view class="weui-label">姓名</view>
    </view>
    <view class="weui-cell__bd">
      <input class="weui-input" placeholder="请输入姓名" bindchange="changeName"/>
```

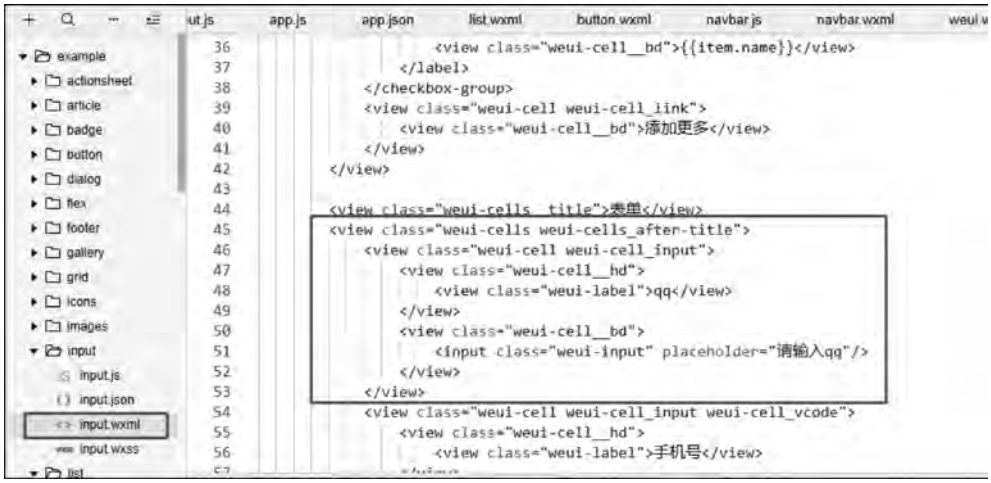

图 3-25　input. wxml 中样式对应的代码

```
    </view>
  </view>
</view>
```

2. 注册页面的功能实现

注册页面的功能实现需要完善 register. js 中的代码，代码如下：

```
Page({
  data: {
    name:''
  },
  changeName:function(e){
    this.setData({
      name: e.detail.value
    })
  }
})
```

其他注册信息的输入框与姓名一样，分别加入 wxml 代码，并在 data 数组中加入对应的变量，将对应的 bindchange()函数进行修改即可。

除了输入框外，最后还有一个“提交”按钮，在 WeUI 样式库中的表单下的 button 中找到对应的 button 样式，如图 3-26 所示。

然后在 register. wxml 文件的最后加上一段 button 的代码，具体代码如下：

```
<view class = "page__bd page__bd_spacing submit">
  <button class = "weui - btn" type = "primary">提交</button>
</view>
```

其中，第一个< view >的 class 类的最后新加一个 submit 子类，并在 wxss 文件中写 submit 子类样式的相关属性，主要是为了调整“提交”按钮的样式。如果 margin 后面只有两个参数，

图 3-26　WeUI 样式库中对应的 button 样式

第一个表示 top 和 bottom，第二个表示 left 和 right。“margin：0 auto”表示上下边界为 0，左右则根据宽度自适应相同值（即居中）；padding-top 的作用是使 button 与 input 之间有一定距离，而不是紧紧连接在一起。设置 width 为屏幕宽度的 90%。具体代码如下：

```
.submit{
  margin: 0 auto;
  padding-top: 15px;
  width: 90%;
}
```

“提交”按钮绑定的事件处理函数 bindSubmit()，主要是向后台发送用户注册信息，这为后台提供了一个 API 接口用于将注册信息存入后台数据库。请求成功后，跳转至 index 页面，具体代码如下：

```
bindSubmit:function (e) {
  wx.request({
    url:'http://zjgsujiaoxue.applinzi.com/index.php/Api/User/register_by_openid',
      data: {
        openid: wx.getStorageSync('jiaoxue_OPENID'),
        globalData:JSON.stringify(app.globalData.userInfo),
        name:this.data.name,
        tel:this.data.tel,
        school:this.data.school,
```

```
          num:this.data.num,
          enter_year:this.data.year
        },
        success: res => {
          if (res.data.is_register) {
            wx.redirectTo({
              url:'../index/index',
            })
          }
        },
        fail: res => {
      },
    })
},
```

3.3 “我的”页面

用户在注册页面填入注册信息后，单击“提交”按钮，完成豆豆云的注册。然后跳转至index页面，这里需要新建一个“我的”页面，用于用户查看注册信息，本节主要讲解如何开发“我的”页面。

3.3.1 “我的”页面知识点讲解

“我的”页面主要新增了两个知识点：微信小程序媒体组件image的属性和wxss属性。

1. image属性

image组件的属性详见表3-6。

表3-6　image组件的属性

属　性　名	类　　型	说　　明
src	string	图片资源地址
mode	string	图片裁剪、缩放的模式
binderror	HandleEvent	当错误发生时，发布到AppService的事件名，事件对象event.detail = {errMsg: 'something wrong'}
bindload	HandleEvent	当图片载入完毕时，发布到AppService的事件名，事件对象event.detail = {height:'图片高度px', width:'图片宽度px'}

注：image组件默认宽度为300px、高度为225px。

图3-27中image组件用到了三目运算作为判断。三目运算符定义为：<表达式1>?<表达式2>:<表达式3>。其含义是：先求表达式1的值，如果为真，则执行表达式2，并返回表达式2的结果；如果表达式1的值为假，则执行表达式3，并返回表达式3的结果。

试验中的image链接语句src="{{userInfo.head_img1? userInfo.head_img:'/images/

```
<view class="weui-cells weui-cells_after-title">
  <view class="weui-cell weui-cell_access" hover-class="weui-cell_active"
bindtap='choseImage'>
    <view class="weui-cell__bd">头像</view>
      <view class="zan-cell__ft weui-cell__ft_in-access ">
        <image class="head_img" src="{{userInfo.head_img?
userInfo.head_img:'/images/default_head_circle.png'}}"></image>
      </view>
  </view>
  <view class="weui-cell weui-cell_access " hover-class="weui-cell_active"
bindtap='bindName'>
    <view class="weui-cell__bd ">姓名</view>
    <view class="weui-cell__ft weui-cell__ft_in-access ">{{userInfo.name}}</view>
  </view>
  <view class="weui-cell weui-cell_access " hover-class="weui-cell_active"
bindtap='bindTel'>
    <view class="weui-cell__bd ">手机号</view>
    <view class="weui-cell__ft weui-cell__ft_in-access ">{{userInfo.tel}}</view>
  </view>
  <view class="weui-cell weui-cell_access " hover-class="weui-cell_active"
bindtap='bindSex'>
    <view class="weui-cell__bd ">性别</view>
```

图 3-27　images 组件

default_head_circle. png'}} 是三目运算符。首先判断 Storage 当中是否获取到 userInfo. head_img，如图 3-28 所示。

图 3-28　userinfo 中头像信息

如果 Storage 中获取到 userInfo. head_img，则图片资源地址为 userInfo. head_img，反之则为 image 文件中的 default_head_circle. png 图片。

2. wxss 属性

rpx(responsive pixel)：可以根据屏幕宽度进行自适应调节。规定屏幕宽为 750rpx。如在 iPhone6 上，屏幕宽度为 375px，共有 750 个物理像素，则 750rpx = 375px = 750 物理像素，1rpx = 0. 5px = 1 物理像素。设备对应的单位换算详见表 3-7。

表 3-7　设备对应的单位换算

设　备	rpx 换算 px(屏幕宽度/750)	px 换算 rpx(750/屏幕宽度)
iPhone5	1rpx=0. 42px	1px=2. 34rpx
iPhone6	1rpx=0. 5px	1px=2rpx
iPhone6 Plus	1rpx=0. 552px	1px=1. 81rpx

建议：开发微信小程序时设计师可以用 iPhone6 作为视觉稿的标准。

注意：在较小的屏幕上不可避免地会有一些毛刺，请在开发时尽量避免这种情况。

```
.head_img {
height: 120rpx;
width: 120rpx;
border-radius: 50%;
}
.weui-cell__ft {
color: #000;
}
```

上述代码中 height 为图片的高度，width 为图片的宽度，border-radius 为圆角的角度，为图片添加圆角边框，例如 border-radius：50%，就是以百分比定义圆角的形状。

3.3.2　“我的”页面实现

右击 pages，在弹出的快捷菜单中选择“新建目录”命令，新建目录并命名为 my。右击 my 目录，在弹出的快捷菜单中选择“新建 Page”命令，新建 Page 并命名为 myinfo。

“我的”页面的实现与注册页面基本相同。其中“我的”页面的效果如图 3-29 所示。

首先就是在 WeUI 样式库中找到对应的样式，查看 WeUI 中 list 样式，发现要找的是“带说明带跳转的列表项”，如图 3-30 所示。myinfo.wxml 文件中的代码如下：

图 3-29　“我的”页面效果

图 3-30　WeUI 样式库中对应的 list 样式

```
<view class="weui-cells weui-cells_after-title">
  <navigator url="" class="weui-cell weui-cell_access" hover-class="weui-cell_
active">
    <view class="weui-cell__bd">头像</view>
    <view class="weui-cell__ft weui-cell__ft_in-access">
      <image class="head_img" src="{{userinfo.head_img?userinfo.head_img:'/images/
default_head_circle.png'}}">
      </image>
    </view>
  </navigator>
  <navigator url="" class="weui-cell weui-cell_access" hover-class="weui-cell_
active">
    <view class="weui-cell__bd">姓名</view>
    <view class="weui-cell__ft weui-cell__ft_in-access">{{userinfo.name}}</view>
  </navigator>
  <navigator url="" class="weui-cell weui-cell_access" hover-class="weui-cell_
active">
    <view class="weui-cell__bd">手机号</view>
    <view class="weui-cell__ft weui-cell__ft_in-access">{{userinfo.tel}}</view>
  </navigator>
  <navigator url="" class="weui-cell weui-cell_access" hover-class="weui-cell_
active">
    <view class="weui-cell__bd">性别</view>
    <view class="weui-cell__ft weui-cell__ft_in-access">{{userinfo.sex}}</view>
  </navigator>
  <navigator url="" class="weui-cell weui-cell_access" hover-class="weui-cell_
active">
    <view class="weui-cell__bd">学校</view>
    <view class="weui-cell__ft weui-cell__ft_in-access">{{userinfo.school}}</view>
  </navigator>
  <navigator url="" class="weui-cell weui-cell_access" hover-class="weui-cell_
active">
    <view class="weui-cell__bd">学号</view>
    <view class="weui-cell__ft weui-cell__ft_in-access">{{userinfo.number}}</view>
  </navigator>
  <navigator url="" class="weui-cell weui-cell_access" hover-class="weui-cell_
active">
    <view class="weui-cell__bd">入学年份</view>
    <view class="weui-cell__ft weui-cell__ft_in-access">{{userinfo.enter_year}}
</view>
  </navigator>
</view>
```

其中，userinfo 的值是通过向后台访问请求，获取到的用户信息，并保存在本地，然后从本地读取出来进行赋值。该请求的代码写在 app.js 中，具体代码如下：

```
wx.request({
  url:'https://zjgsujiaoxue.applinzi.com/index.php/Api/User/getInfo',
  data: {
    'openid': res.data.openid,
  },
  success:function (res1) {
```

```
      wx.setStorageSync('userInfo', res1.data.data)
    },
})
```

在 myinfo.js 文件的 data 数组中定义变量 userinfo，并在 onLoad()函数中对 userinfo 变量进行赋值，具体代码如下：

```
Page({

/**
 * 页面的初始数据
 */
data: {
userinfo:{ }
},

/**
 * 生命周期函数--监听页面加载
 */
onLoad:function (options) {
this.setData({
userinfo: wx.getStorageSync('userInfo')
})
}
```

编译后发现头像显示过大，如图 3-31 所示。

图 3-31　头像显示过大

因此需要在媒体组件 image 中自定义类 head_img，调整图片大小。其中 myinfo.wxss 文件的代码如下：

```
.head_img{
height: 120rpx;
width: 120rpx;
border-radius: 50%;
}
```

到这里，“我的”页面就能正常显示了。

3.4 作业思考

一、讨论题

1. 讨论对小程序登录流程的理解。
2. 如何理解数据缓存中同步与异步缓存的区别？
3. 如何快速找到并使用 WeUI 基础样式库中自己需要的样式？
4. 样式中 margin 属性值为 0 auto 是什么意思？
5. bindchange 与 bindtap 有什么区别？
6. 新建 tabBar 之后，register 页面中页面跳转的逻辑是否需要修改？
7. 如何修改图片的大小和形状？

二、单选题

1. wx.login()有(　　)属性。
   A. success、fail、timeout、complete
   B. success、fail、data、complete
   C. success、fail、timeout、data
   D. success、fail、url、data
2. 以下关于 wx.showModal()的说法错误的是(　　)。
   A. Title 是模态对话框的标题
   B. Content 是模态对话框的内容
   C. showCancel 是否取消模态对话框
   D. cancelText 是“取消”按钮的文字
3. 以下关于 wx.request()的说法不正确的是(　　)。
   A. URL 是开发者服务器的接口地址
   B. data 是请求的参数
   C. complete()是调用结束的回调函数(只有调用成功才会执行)
   D. dataType 默认值是 json
4. 关于以下 API 请求的说法错误的是(　　)。

```
wx.request({
```

```
    url: 'https://zjgsujiaoxue.applinzi.com/index.php/Api/Weixin/code_to_openidv2',
    data: {
      questionA: right
    },
    success: function(res1) {
      console.log('http 返回值', res1)
    },
    fail: function(res2) {
      console.log('http 返回值', res2)
    }
  })
```

A. questionA:right 表示向后台传送字符串'right'

B. console.log('http 返回值',res1)表示请求成功打印后台返回值

C. console.log('http 返回值',res2)表示请求失败打印后台返回值

D. 该请求的 HTTP 请求方式是 POST

5. 当 wxml 的 input 组件通过 bindchange 事件绑定了 js 的 changname: function(e)函数时,打印 input 组件中改变的值,则使用(　　)。

A. console.log(e.detail.value)

B. console.log(e.detail.input)

C. console.log(e.value)

D. console.log(e.input)

6. 以下关于 image 组件的属性的说法中,(　　)是错误的。

A. src: 图片的资源地址

B. mode: 图片裁剪、缩放的模式

C. binderror:当没有错误发生时,发布到 AppService 的事件名,事件对象 event.detail={ errMsg: 'something wrong' }

D. bindload: 当文档载入完毕时,发布到 AppService 的事件名,事件对象 event.detail={height:'图片高度 px', width:'图片宽度 px'}

7. 关于三目运算符的定义:<表达式 1>?<表达式 2>:<表达式 3>,以下表述正确的是(　　)。

A. 先求表达式 1 的值,如果为真,则执行表达式 2,并返回表达式 2 的结果

B. 先求表达式 1 的值,如果为真,则执行表达式 3,并返回表达式 3 的结果

C. 如果表达式 1 的值为假,则执行表达式 2,并返回表达式 2 的结果

D. 如果表达式 1 的值为真,则执行表达式 2 和 3,并返回表达式 2 和 3 的结果

8. 以下关于 rpx 的说法正确的是(　　)。

A. iPhone6 上 1rpx = 0.5px = 1 物理像素

B. iPhone5 上 1rpx = 0.5px = 1 物理像素

C. iPhone6 上 1rpx = 0.42px = 1 物理像素

D. iPhone6 Plus 上 1rpx = 0.5px = 1 物理像素

9. 以下关于 border-radius 的说法正确的是(　　)。

A. 为图片添加边框　　B. 为图片添加圆角边框

C. 为文字添加圆角边框　　D. 为图片改变边框大小

10. 以下关于数据缓存 API 函数类型的说法不正确的是(　　)。

A. wx.setStorage(Object object)实现数据的异步存储

B. wx.setStorage(Object object)实现数据的同步存储

C. wx.getStorage(Object object)实现数据的异步获取

D. wx.getStorageInfo(Object object)实现存储信息的异步获取