

# 第七章 小程序集成开发环境实验

本章主要完成小程序的集成开发环境实验。要求下载安装并熟练使用微信小程序开发环境，能够理解微信小程序的结构和原理，熟悉微信小程序 WXML 页面基本语法以及微信小程序的开发语言 Javascript。对小程序的进一步了解，前端开发可以参考文献[11]，后端以及云开发可以参考文献[12]。

## 第一节 小程序注册

进入微信公众平台 <https://mp.weixin.qq.com/>使用自己的 email 注册微信小程序，注意该页面上面有“立即注册”的链接，点击该链接则可以看到注册各种服务，包括订阅号，服务号，小程序和企业微信等，如图 7.1 所示。



图 7.1 微信小程序的注册

注册的时候使用电子邮件认证，用户打开认证时平台发送的电子邮件，并使用微信软件扫码邮件中给出的二维码，以后就可以使用这个微信账号直接登录微信公众平台实现对小程序的管理了，同时也可以用来扫码微信小程序开发工具进行小程序的开发工作。注册成功以后可以登录微信公众平台查看自己的 appID 等微信小程序开发信息，在开发小程序的时候需要使用这个 appID 来标识该微信小程序。图 7.2 是注册小程序以后给出的小程序账户信息，可以查看到 AppID，注册时的登录邮箱等。



图 7.2 微信小程序账户信息

另外微信公众平台还可以实现对小程序的管理，例如版本管理，统计、基本设置以及公众号关联等等操作。小程序的具体生命周期管理如下：

- (1)注册：在微信公众平台注册小程序，注册的时候需要电子邮件认证，认证的时候需要用户使用自己的微信账号扫码，以后可以使用该微信账号扫码微信公众平台管理开发的小程序；
- (2)小程序信息完善：填写小程序基本信息，包括名称、头像、介绍及服务范围等。需要注意的

是小程序名称很重要，用户使用该名称实现对小程序的搜索查找，用户每次修改名称都要经过审查，而且修改次数受到限制；

(3)开发小程序：用户需要下载微信小程序开发者工具、参考开发文档对小程序进行开发和调试。第一次打开微信小程序开发工具的时候，需要使用管理微信公众平台的账号来扫码微信小程序开发工具，进而进行小程序开发。

(4)提交审核和发布:完成小程序开发后，提交代码至微信小程序团队审核，审核通过后即可发布（公测期间不能发布）。

## 第二节 小程序开发工具安装

访问目录 <https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>，下载最新的微信小程序开发工具并进行安装，安装成功后运行该开发工具，可以快速浏览开发工具的菜单，了解其基本功能，特别要说明的是 ctrl-z 是编辑器的回退，ctrl-y 是前行，这样方便删除后的快速恢复。

### 一、小程序运行示例

访问 <https://developers.weixin.qq.com/miniprogram/dev/framework/quickstart/#小程序简介>。该链接有小程序体验二维码，如图 3 所示。微信扫码后可以体验小程序各种组件的功能，也能访问相应的源码，可以直接复制用于自己编写小程序。这方便用户在体验小程序组件的同时，可以查看具体实现代码。



图 7.3 小程序体验二维码

### 二、生成第一个小程序

打开微信小程序开发工具需要使用注册时候的微信账户进行微信扫码，如图 7.4 所示，扫码成功后即可打开小程序开发工具界面。此后对开发的小程序上传和发布都可以通过微信扫码登录微信公众平台 <https://mp.weixin.qq.com/>，进而对小程序直接进行管理。



图 7.4 微信开发工具的扫码登录

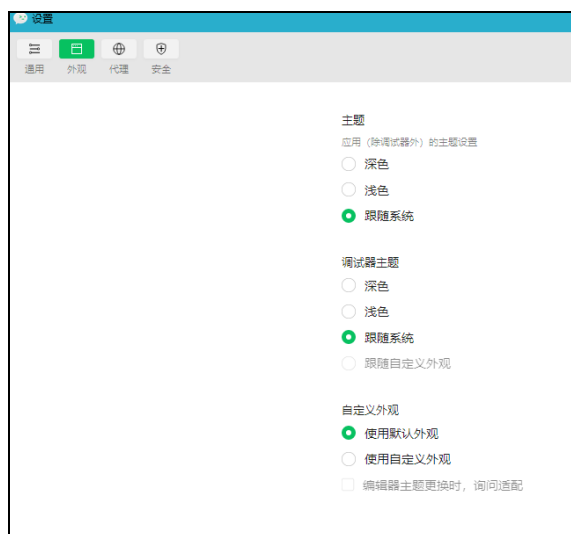


图 7.5 微信开发工具的背景设置

第一次运行微信开发工具界面时，点击右上角设置按钮，可以选择该开发工具的外观菜单，也就是可以修改开发工具的颜色，比如选择“浅色”或者“跟随系统”，开发工具界面的显示将和一般

软件的界面颜色一样，如图 7.5 所示。

微信扫码图 7.4 所示的二维码成功后，就相当于使用该微信账号登录了微信小程序开发工具，以后使用该工具不用再扫码了，然后就进入了“小程序项目”，如图 7.6 所示。



图 7.6 小程序项目的装载或者创建

进入小程序项目界面以后，需要到微信公众平台中，将自己的 AppID 拷贝粘贴到图 7.6 中的 AppID 对应的文本框中。需要说明的是扫码开发工具的微信账号必须和扫码微信公众平台的微信账号是同一个账号，否则无法将 AppID 拷贝到上面的 AppID 文本框中。这个原因就是开发工具开发的小程序上传到微信公众平台的时候，是通过 AppID 识别的。如果使用不同的微信账号扫码登录，则无法实现公众平台对上传的小程序进行识别和管理。

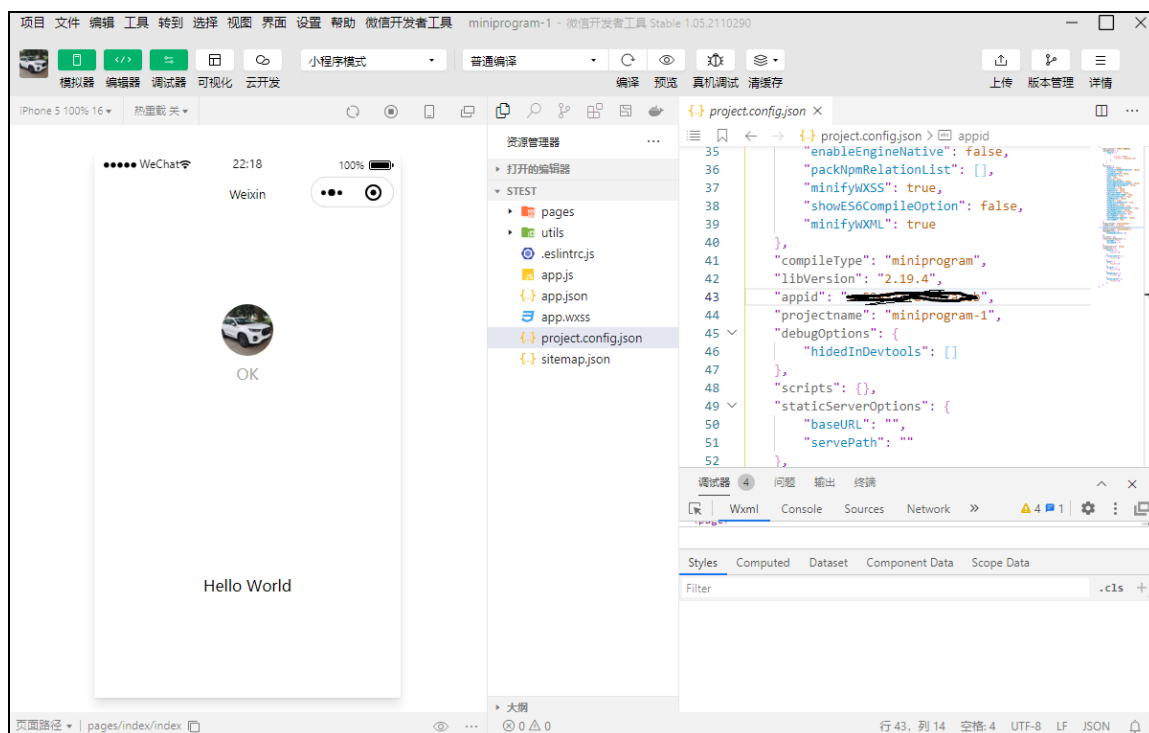


图 7.7 自动生成一个不使用云服务的小程序

第一次开发小程序，可以选择“不使用云服务”，所谓云服务就是微信公众平台提供的远程数据库等云端服务。选择左侧列表中的“小程序”，则开发工具自动创建一些文件夹和文件，如图 7.7 所示。这是开发工具自动生成的小程序示例，可以“编译”，“预览”和“真机调试”运行第一个小程序“hello world”，用户可以在此示例项目中添加自己的页面，开发自己的小程序。

点击开发工具上面的“预览”以及“真机调试”后，会显示二维码，微信扫描这个二维码以后，就会在手机微信上运行该小程序，但是只能自己以及项目成员看到这个小程序，如果要上线，必须提交审核通过

小程序开发完成以后，可以点击开发工具右上角的“上传”。上传成功以后，打开微信公众平台，点击该页面的版本管理，就可以看到上传的小程序了。当然扫描的微信账号必须和扫描开发工具的微信账号一致，否则无法看到上传的小程序。

### 第三节 小程序目录结构

打开微信小程序集成开发环境，创建一个小程序项目，就会发现有三个 `app` 文件和 `pages` 文件夹。小程序的三个 `app` 文件分别是 `app.js`，`app.json` 和 `app.wxss`，它们放在项目的根目录下，各自功能如表 7.1 所示，

表 7.1 小程序主体 `app` 文件

文件	必需	作用
<code>app.js</code>	是	小程序整体逻辑定义
<code>app.json</code>	是	小程序公共配置
<code>app.wxss</code>	否	小程序公共样式表

在 `page` 文件夹下，是小程序的页面，每一个小程序页面由四个文件组成，其扩展名分别是 `js`，`wxml`，`json` 和 `wxss`，表示四个文件类型，其功能如表 7.2 所示。

表 7.2 小程序页面的四个文件的功能

文件类型	必需	作用
<code>js</code>	是	页面逻辑
<code>wxml</code>	是	页面结构
<code>json</code>	否	页面配置
<code>wxss</code>	否	页面样式表

小程序页面如果有公共页面可以使用 `#include` 包含进来。页面之间共享数据可以使用 `wx.getStorageSync("userName")` 获得数据，需要保存数据时候，类似使用 `wx.setStorageSync("userName", returnName)` 语句实现。为了防止持久保存占用太多内存，可以使用 `wx.clearStorage()` 清除所有数据，也可以具体指出清除哪些数据。这个功能类似 JSP 等 web 编程语言的 `session` 内置对象；`js` 类型的文件负责具体数据处理逻辑的实现，而 `wxml` 具体负责页面呈现，`wxml` 可以将数据传给 `js` 文件来具体进行业务逻辑的运算和处理，而 `js` 处理的数据结果可以传给 `wxml` 实现数据的显示。

#### 一、项目配置文件

使用小程序开发工具的时候，用户可能做一些个性化配置，例如开发工具的界面颜色、编译配置等等。为了记录这些配置，开发工具在每一个项目的根目录都会生成一个 `project.config.json`，所做的工具配置都会写入到这个文件，用户重新安装工具或者更换电脑工作时，只要载入同一个项目的代码包，开发者工具就自动会恢复这些个性化配置。

这个文档中比较重要的就是 `appid` 的配置，如果用户打开别人的小程序代码就只能看，无法编译，因此只要此处修改为自己的 `appid` 即拥有了所有的权限，另外 `projectname` 可以修改项目名称。这些配置可以在开发工具菜单“设置”二级菜单“项目设置”修改，如图 7.8 所示。如果修改这些设置，修改的结果会自动保存到 `project.config.json` 文件中。



图 7.8 小程序项目的配置界面

## 二、小程序的全局内容定义

小程序是一个软件，运行的时候诸如小程序名称，小程序的全局变量，小程序的背景颜色，控件风格，小程序有哪些页面等等总要要进行设置，这种设置就放在小程序项目列表中的三个文档中，即 app.js; app.json; app.wxss

### 1. app.js 文件

该文件的主要作用就是小程序启动后，提供小程序的应用实例以及小程序关闭，隐藏和显示需要运行的程序。app.js 作为项目的入口文件，用于创建应用程序对象，列表 7.1 中注释了 app.js 中各个函数的功能，比如 onLaunch(), onShow()函数等都是在小程序启动后在各种状态下触发运行实现的一些功能。

列表 7.1 小程序的 app.js 文件主要功能

```
//App 函数是一个全局函数，用于创建应用程序对象
App({ // =====全局数据对象（可以整个应用程序共享）=====
  globalData: { ... },
  //=====应用程序全局方法=====
  method1(p1, p2) { ... },
  method1(p1, p2) { ... },
  //=====生命周期方法=====
  //应用程序启动时触发一次
  onLaunch() { ... },
  //当应用程序进入前台显示状态时触发
  onShow() { ... },
  //当应用程序进入后台状态时触发
  onHide() { ... }
})
##### 小程序-调用公共 js 对象方法/ app.js
### 在小程序中，如果在子页面想调用公共 js 的方法，需先在子页面 js 中先实例化 app:具体过程如下 子页面 js:
//调用公共 js 对象以便调用其方法
var app = getApp();//获取应用实例
app.globalData.userInfo = xxx -->赋值
// console.log(app)//可查看公共 js (app.js) 的方法
```

### 2. app.json 文件

app.json 是对整个小程序的全局配置。在这个文件中对小程序的配置进行说明：(1) 页面的组成；(2) 小程序的窗口颜色设置；(3) 导航条样式的设置；(4) 默认标题的设置等等。另外该文件中不可以添加任何的注释，否则会报错。列表 7.2 是使用云开发由微信小程序开发工具自动创建的一个小程序一个示例。其中，pages 字段用于描述当前小程序所有页面路径，这是为了让微信客户端知道当前小程序页面定义在哪个目录下，也就是小程序所有的页面都需要在 pages 目录下定义，一旦定义完成就会在相应的目录下自动生成四个文档，即以 js、json、xml 和 wxss 为结尾的四个文件类型。

需要注意的是排在 pages 中第一个页面是主页面，小程序启动后最先运行它。window 字段定义小程序所有页面的顶部背景颜色和文字颜色等。

列表 7.2 app.json 文件配置例子

```
"cloud": true,
"pages": [
  "pages/main/main",
  "pages/userConsole/userConsole",
  "pages/storageConsole/storageConsole",
  "pages/databaseGuide/databaseGuide",
  "pages/addFunction/addFunction",
  "pages/deployFunctions/deployFunctions",
  "pages/chooseLib/chooseLib"
],
"window": { -----小程序的窗体的基本设置，注意背景颜色，文本风格等等。
  "backgroundColor": "#F6F6F6",
  "backgroundTextStyle": "light",
  "navigationBarBackgroundColor": "#F6F6F6",
  "navigationBarTitleText": "云开发 QuickStart",
  "navigationBarTextStyle": "black"
},
"sitemapLocation": "sitemap78.json"
```

### 3. app.wxss

app.wxss 是对整个小程序的公共样式进行定义，此处定义的样式在所有界面都可以用。这种样式类似 html 中的 css，是对各种控件和框架元素定义大小，颜色位置等。

### 三、微信小程序页面

在小程序中，一个完整的页面是由 js、wxml、wxss 和 json 为扩展名的四个类型的文件组成，而且这四个文件的主文件名必须相同而且位于以主文件名命名的目录下。

js 页面逻辑文件。所有的业务逻辑都要在这里实现，类似 web 编程里面嵌入 JSP 或者 ASP 页面的 Java 或者 VB 代码，编写 js 脚本文件来处理用户的操作。

wxml 结构文件类似 web 页面。web 页面采用的是 html + css + js 这样的组合，html 是用来描述当前这个页面的结构，css 描述的是页面的样式，js 用来处理这个页面和用户的交互。wxml 类似 html 的角色，不同之处在于 html 会用到标签 div, p, span，然后根据这些标签组合出不一样的组件，例如日历、弹窗等等，而小程序的 wxml 用的标签是 view, button, text 等等。小程序封装了这些标签，直接提供了各种组件能力，而且还提供了地图、视频和音频等等组件。

wxss 页面样式文件。wxml 里面的 css 样式，它跟普通 css 文件的区别有 wxml 文件可以导入其他的 wxml 文件。另外，新增了尺寸单位，以适应手机设备屏幕的宽度和设备像素比。wxss 在底层支持新的尺寸单位 rpx(responsive pixel 反应像素)。提供了全局的样式和局部样式，例如写一个 app.wxss 作为全局样式，它作用于当前小程序的所有页面，局部页面样式 page.wxss 仅对当前页面生效。

json 页面配置文件。页面的 json 配置会覆盖 app.json 的配置。

## 第四节 专属小程序快速创建

目前，有大量的互联网公式提供小程序模板服务。小程序模板可以快速创建小程序，提交审核使用同时推广。用户可以自己选择行业模板进行编辑，也就是更改图片和文字说明，联系方式等，更改完成以后一样要经过审核等步骤，互联网企业可以帮助推广等。网站 <https://i.fkw.com/portal.jsp> 提供免费制作生成小程序，学生实习的时候可以选择定制一些行业的小程序。

## 第五节 综合实验七

## 一、实验目的和要求

1. 掌握微信小程序开发工具的使用；
2. 掌握微信小程序的目录结构和框架等基本思想；
3. 掌握微信公众平台对小程序生命周期的管理；
4. 掌握专属小程序的快速创建。

## 二、实验环境与工具软件

1. 微信小程序开发工具；
2. 网络环境
3. 微信运行环境。

## 三、实验内容和步骤

下载微信小程序开发工具，学习如何使用该集成开发环境的各个菜单的功能，包括模拟器，编辑器，调试器以及云开发的使用。能够编译，预览以及真机调试小程序等；分别选择“不使用云服务”和“小程序云开发”生成小程序示例并运行、查看、分析和理解各个页面内容以及它们的关系；上传小程序示例，使用微信公众平台实现对上传的小程序进行测试发布；了解小程序的命名，说明和提交审查等；访问网站 <https://i.fkw.com/portal.jsp>，学习快速创建小程序。

1. 小程序用户注册和开发工具安装
  - (1) 访问 <https://mp.weixin.qq.com/>;
  - (2) 选择账号分类中的“小程序”；
  - (3) 阅读该页面，然后点击“前往注册”完成小程序注册；
  - (4) 进入小程序管理平台，逐项理解基本设置，第三方设置以及版本管理等管理平台的各项功能。
2. 微信小程序开发工具的安装和使用
  - (1) 访问 <https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>;
  - (2) 下载安装开发工具；
  - (3) 熟悉该开发工具菜单中的各项功能；
  - (4) 选择项目中的新建项目，然后选择不使用云开发生成小程序；
  - (5) 分析小程序组成部分，理解它们的目录结构；
  - (6) 在 app.json 中的 pages 里面增加“pages/test/test”然后编译，查看结果。
  - (7) 在 test.wxml 中增加一些组件，查看运行结果；
  - (8) 选择预览和真机调试，查看运行结果。
3. 小程序商城快速创建
  - (1) 访问网站 <https://i.fkw.com/portal.jsp>;
  - (2) 注册用户；
  - (3) 选择小程序模板；
  - (4) 编辑修改，重建自己的小程序；
  - (5) 上传提交审核。

## 四、实验报告(共5分)

1. 小程序用户注册步骤与说明 (1分)
2. 微信小程序开发工具的安装和使用说明 (2分)
3. 编写一个微信小程序,手机运行后显示“我的小程序” (1分)
4. 说明微信公众平台对小程序的管理 (1分)