# 微信小程序

董业 201400130044

QQ:1342156335

**1. 获取微信小程序的 AppID**

登录 [https://mp.weixin.qq.com](https://mp.weixin.qq.com/?t=20161122) ，就可以在网站的“设置”-“开发者设置”中，查看到微信小程序的 AppID 了，注意不可直接使用服务号或订阅号的 AppID 。



**注意：如果要以非管理员微信号在手机上体验该小程序，那么我们还需要操作“绑定开发者”。即在“用户身份”-“开发者”模块，绑定上需要体验该小程序的微信号。本教程默认注册帐号、体验都是使用管理员微信号。**

### 2. 创建项目

我们需要通过[开发者工具](https://mp.weixin.qq.com/debug/wxadoc/dev/devtools/devtools.html?t=20161122)，来完成小程序创建和代码编辑。

开发者工具安装完成后，打开并使用微信扫码登录。选择创建“项目”，填入上文获取到的 AppID ，设置一个本地项目的名称（非小程序名称），比如“我的第一个项目”，并选择一个本地的文件夹作为代码存储的目录，点击“新建项目”就可以了。



项目创建成功后，我们就可以点击该项目，进入并看到完整的开发者工具界面，点击左侧导航，在“编辑”里可以查看和编辑我们的代码，在“调试”里可以测试代码并模拟小程序在微信客户端效果，在“项目”里可以发送到手机里预览实际效果。

如果没有appID，也可以进行本地调试，但是部分功能受限。

为方便初学者了解微信小程序的基本代码结构，在创建过程中，如果选择的本地文件夹是个空文件夹，开发者工具会提示，是否需要创建一个 quick start 项目。选择“是”，开发者工具会帮助我们在开发目录里生成一个简单的 demo。

3. 编写代码

创建小程序实例

点击开发者工具左侧导航的“编辑”，我们可以看到这个项目，已经初始化并包含了一些简单的代码文件。最关键也是必不可少的，是 app.js、app.json、app.wxss 这三个。其中，.js后缀的是脚本文件，.json后缀的文件是配置文件，.wxss后缀的是样式表文件。微信小程序会读取这些文件，并生成小程序实例。

下面我们简单了解这三个文件的功能，方便修改以及从头开发自己的微信小程序。

app.js是小程序的脚本代码。我们可以在这个文件中监听并处理小程序的生命周期函数、声明全局变量。调用框架提供的丰富的 API，如本例的同步存储及同步读取本地数据。想了解更多可用 API，可参考 API 文档

//app.js

App({

onLaunch: function () {

//调用API从本地缓存中获取数据

var logs = wx.getStorageSync('logs') || []

logs.unshift(Date.now())

wx.setStorageSync('logs', logs)

},

getUserInfo:function(cb){

var that = this

if(this.globalData.userInfo){

typeof cb == "function" && cb(this.globalData.userInfo)

}else{

//调用登录接口

wx.login({

success: function () {

wx.getUserInfo({

success: function (res) {

that.globalData.userInfo = res.userInfo

typeof cb == "function" && cb(that.globalData.userInfo)

}

})

}

})

}

},

globalData:{

userInfo:null

}

})

app.json 是对整个小程序的全局配置。我们可以在这个文件中配置小程序是由哪些页面组成，配置小程序的窗口背景色，配置导航条样式，配置默认标题。注意该文件不可添加任何注释。更多可配置项可参考配置详解。

{

"pages":[

"pages/index/index",

"pages/logs/logs"

],

"window":{

"backgroundTextStyle":"light",

"navigationBarBackgroundColor": "#fff",

"navigationBarTitleText": "WeChat",

"navigationBarTextStyle":"black"

}

}

app.wxss 是整个小程序的公共样式表。我们可以在页面组件的 class 属性上直接使用 app.wxss 中声明的样式规则。

/\*\*app.wxss\*\*/

.container {

height: 100%;

display: flex;

flex-direction: column;

align-items: center;

justify-content: space-between;

padding: 200rpx 0;

box-sizing: border-box;

}

#### 创建页面

在这个教程里，我们有两个页面，index 页面和 logs 页面，即欢迎页和小程序启动日志的展示页，他们都在 pages 目录下。微信小程序中的每一个页面的【路径+页面名】都需要写在 app.json 的 pages 中，且 pages 中的第一个页面是小程序的首页。

每一个[小程序页面](https://mp.weixin.qq.com/debug/wxadoc/dev/framework/app-service/page.html?t=20161122)是由同路径下同名的四个不同后缀文件的组成，如：index.js、index.wxml、index.wxss、index.json。.js后缀的文件是脚本文件，.json后缀的文件是配置文件，.wxss后缀的是样式表文件，.wxml后缀的文件是页面结构文件。

index.wxml 是页面的结构文件：

<!--index.wxml-->

<view class="container">

<view bindtap="bindViewTap" class="userinfo">

<image class="userinfo-avatar" src="{{userInfo.avatarUrl}}" background-size="cover"></image>

<text class="userinfo-nickname">{{userInfo.nickName}}</text>

</view>

<view class="usermotto">

<text class="user-motto">{{motto}}</text>

</view>

</view>

本例中使用了[<view/>](https://mp.weixin.qq.com/debug/wxadoc/dev/component/view.html?t=20161122)、[<image/>](https://mp.weixin.qq.com/debug/wxadoc/dev/component/image.html?t=20161122)、[<text/>](https://mp.weixin.qq.com/debug/wxadoc/dev/component/text.html?t=20161122)来搭建页面结构，绑定数据和交互处理函数。

index.js 是页面的脚本文件，在这个文件中我们可以监听并处理页面的生命周期函数、获取小程序实例，声明并处理数据，响应页面交互事件等。

define("pages/index/index.js", function(require, module, exports, window,document,frames,self,location,navigator,localStorage,history,Caches,screen,alert,confirm,prompt,XMLHttpRequest,WebSocket,webkit,WeixinJSCore,WeixinJSBridge,Reporter){ 'use strict';

//index.js

//获取应用实例

var app = getApp();

Page({

data: {

circleList: [], //圆点数组

awardList: [], //奖品数组

colorCircleFirst: '#FFDF2F', //圆点颜色1

colorCircleSecond: '#FE4D32', //圆点颜色2

colorAwardDefault: '#F5F0FC', //奖品默认颜色

colorAwardSelect: '#ffe400', //奖品选中颜色

indexSelect: 0, //被选中的奖品index

isRunning: false, //是否正在抽奖

imageAward: ['../../images/1.jpg', '../../images/2.jpg', '../../images/3.jpg', '../../images/4.jpg', '../../images/5.jpg', '../../images/6.jpg', '../../images/7.jpg', '../../images/8.jpg'] },

onLoad: function onLoad() {

var \_this = this;

//圆点设置

var leftCircle = 7.5;

var topCircle = 7.5;

var circleList = [];

for (var i = 0; i < 24; i++) {

if (i == 0) {

topCircle = 15;

leftCircle = 15;

} else if (i < 6) {

topCircle = 7.5;

leftCircle = leftCircle + 102.5;

} else if (i == 6) {

topCircle = 15;

leftCircle = 620;

} else if (i < 12) {

topCircle = topCircle + 94;

leftCircle = 620;

} else if (i == 12) {

topCircle = 565;

leftCircle = 620;

} else if (i < 18) {

topCircle = 570;

leftCircle = leftCircle - 102.5;

} else if (i == 18) {

topCircle = 565;

leftCircle = 15;

} else if (i < 24) {

topCircle = topCircle - 94;

leftCircle = 7.5;

} else {

return;

}

circleList.push({ topCircle: topCircle, leftCircle: leftCircle });

}

this.setData({

circleList: circleList

});

//圆点闪烁

setInterval(function () {

if (\_this.data.colorCircleFirst == '#FFDF2F') {

\_this.setData({

colorCircleFirst: '#FE4D32',

colorCircleSecond: '#FFDF2F'

});

} else {

\_this.setData({

colorCircleFirst: '#FFDF2F',

colorCircleSecond: '#FE4D32'

});

}

}, 500);

//奖品item设置

var awardList = [];

//间距,怎么顺眼怎么设置吧.

var topAward = 25;

var leftAward = 25;

for (var j = 0; j < 8; j++) {

if (j == 0) {

topAward = 25;

leftAward = 25;

} else if (j < 3) {

topAward = topAward;

//166.6666是宽.15是间距.下同

leftAward = leftAward + 166.6666 + 15;

} else if (j < 5) {

leftAward = leftAward;

//150是高,15是间距,下同

topAward = topAward + 150 + 15;

} else if (j < 7) {

leftAward = leftAward - 166.6666 - 15;

topAward = topAward;

} else if (j < 8) {

leftAward = leftAward;

topAward = topAward - 150 - 15;

}

var imageAward = this.data.imageAward[j];

awardList.push({ topAward: topAward, leftAward: leftAward, imageAward: imageAward });

}

this.setData({

awardList: awardList

});

},

//开始游戏

startGame: function startGame() {

if (this.data.isRunning) return;

this.setData({

isRunning: true

});

var \_this = this;

var indexSelect = 0;

var i = 0;

var timer = setInterval(function () {

indexSelect++;

//这里我只是简单粗暴用y=30\*x+200函数做的处理.可根据自己的需求改变转盘速度

i += 30;

if (i > 1000) {

//去除循环

clearInterval(timer);

//获奖提示

wx.showModal({

title: '恭喜您',

content: '获得了第' + (\_this.data.indexSelect + 1) + "个优惠券",

showCancel: false, //去掉取消按钮

success: function success(res) {

if (res.confirm) {

\_this.setData({

isRunning: false

});

}

}

});

}

indexSelect = indexSelect % 8;

\_this.setData({

indexSelect: indexSelect

});

}, 200 + i);

}

});

页面的样式表是非必要的。当有页面样式表时，页面的样式表中的样式规则会层叠覆盖 app.wxss 中的样式规则。如果不指定页面的样式表，也可以在页面的结构文件中直接使用 app.wxss 中指定的样式规则。

index.json 是页面的配置文件：

页面的配置文件是非必要的。当有页面的配置文件时，配置项在该页面会覆盖 app.json 的 window 中相同的配置项。如果没有指定的页面配置文件，则在该页面直接使用 app.json 中的默认配置。

logs 的页面结构

<!--logs.wxml-->

<view class="container log-list">

<block wx:for="{{logs}}" wx:for-item="log">

<text class="log-item">{{index + 1}}. {{log}}</text>

</block>

</view>

logs 页面使用 [<block/>](https://mp.weixin.qq.com/debug/wxadoc/dev/framework/view/wxml/list.html?t=20161122%20#block-wxfor) 控制标签来组织代码，在 <block/> 上使用 [wx:for](https://mp.weixin.qq.com/debug/wxadoc/dev/framework/view/wxml/list.html?t=20161122%20#block-wxfor) 绑定 logs 数据，并将 logs 数据循环展开节点

varutil = require('../../utils/util.js')

Page({

data: {

logs: []

},

onLoad: function () {

this.setData({

logs: (wx.getStorageSync('logs') || []).map(function (log) {

return util.formatTime(new Date(log))

})

})

}

})

详细的源代码查看源代码文件夹。

微信小程序运行结果如下：



预览在手机上的效果。

这次写的这个小程序是可以模拟抽奖。由于没有权限，所以只能在本地调试，无法在手机上调试。