

【微信小程序】从入门到放弃

前言

关于**微信**小程序是什么，能做什么的问题，草民在此不在罗列了，随着小程序的天天刷屏，想必您也是来吃一些干货，本篇博文和大家走进微信小程序的从入门到放弃~

微信小程序开放功能

草民看过很多的文档，微信的官方文档写的着实不错了，草民真是发自内心夸赞一下腾讯的这个能力，大家可以前往阅读，下面通过一张图来总览一下小程序开放的功能：

- ▶ 视图容器：视图(View)、滚动视图、Swiper
- ▶ 基础内容：图标、文本、进度条
- ▶ 表单组件：按钮、表单等等
- ▶ 操作反馈
- ▶ 导航
- ▶ 媒体组建：音频、图片、视频。
- ▶ 地图
- ▶ 画布
- ▶ 文件操作能力
- ▶ 网络：上传下载能力、WebSocket
- ▶ 数据：数据缓存能力
- ▶ 位置：获取位置、查看位置
- ▶ 设备：网络状态、系统信息、重力感应、罗盘
- ▶ 界面：设置导航条、导航、动画、绘图等等
- ▶ 开放接口：登录，包括签名加密，用户信息、微信支付、模板消息

功能解读:

微信小程序在 UI 部分沿用了自己 weui 的基础样式, 且由于小程序依托于微信 app, 这样小程序就的很多 api 可以以桥接的方式来调用原声接口, 包括导航、地图、选择图片、IO、网络、登录、支付等功能必然都是通过桥接来实现的, 这也是小程序相比应用号和其他 webapp 的优势~

微信小程序跨平台动态化方案

由于微信小程序源码并未开源, 关于微信小程序采用的动态化原理大家也都在猜测中, 先简单普及一下最新的两种动态化方案:

hybird

采用传统 htm+css+js 的框架进行开发:

优点:

- 1.简单 快速, 模版布局现成, 熟练的 H5 开发工程师都可以胜任
- 2.颠覆性的 **React** 框架的出现之后引入了 Virtual DOM 的概念, 强大的 DOM diff 操作是 UI 更新更加流畅

缺点:

- 1.渲染速度与性能方面的表现依然与原声相差甚远
- 2.从内存方面的考虑, 列表复用问题无法解决

* **React Native** 与 Weex*

优点:

- 1.开发沿用 html 或 xml+css+js 的方式, 但是通过 transform 层将 html 与 css 转换成 Virtual DOM, native 端进行映射渲染, 性能方面大可不必担心。

2.在 native 端植入了 JS 引擎，这样就可以扩展 native api，调用一些系统权限或者扩展功能，例如选择图片。

缺点：

1.虽然沿用了前端传统开发方式，但是现阶段的 html 或 xml 与 css 的标准性是不够的，包括 svg 、css 绘制图标等

2.需要了解部分的 native 知识，对开发效率方面是一个降低

通过上面两种方案的比较，个人还是更倾向于后者的，毕竟性能问题依然是很重要的，最初也猜测微信小程序选择的是后者，毕竟开放功能里面有很多是 native 才拥有的能力。

...

可结论要让大家失望了，微信小程序采用的是前者，有很多人依然不相信，说我强大的腾讯帝国不可能选择落后的方案，来看一张图片：



了解 **Android** 的人都知道上图的意思，不过为了给其他工种程序员讲解，还是简单介绍一下这个功能，在 **android** 的开发者功能里边有一个显示布局边界的设置，这个功能可以把所有 **android** 原声控件的布局边界都显示出来，而上图的 **navigationBar** 与底部的 **bottomBar** 都是原生的，但是中间大

部分内容都是 webview，也证明了之前的观点，微信小程序采用的是 **Hybird** 方式。

很多人要开始吐槽说，说草民你讲这些没用啊，我们又不关心原理，我们只想吃肉，不管是那头猪身上的，有那么一句话，不想当不想当厨子的裁缝不是好司机，想必大家不会想一直站在别人的肩上写代码吧，希望大家能够领悟到草民做这么多铺垫的意义。好吧，话不多说，接下来我们正式开始小程序的讲解。

开发准备

一、下载开发工具

链接地址

二、添加项目



三、填写项目信息



appid 是小程序的，如果没有的话，选择无 APPID，填写完成之后点击添加项目即可~

项目架构

程序主体

一个框架程序主体部分由三个文件组成，必须放在项目的根目录，如下：

文件	必填	作用
<code>app.js</code>	是	小程序逻辑
<code>app.json</code>	是	小程序公共设置
<code>app.wxss</code>	否	小程序公共样式表

1、`app.js`：主要存放一些公共的业务逻辑，或者监听小程序声明周期等操作

注：在任意一个 `page` 的 `js` 中，都可以通过调用 `getApp()` 获取 `app.js` 对象

```
App({
  onLaunch: function() {
    // Do something initial when launch.
  },
  onShow: function() {
    // Do something when show.
  },
  onHide: function() {
    // Do something when hide.
  },
  globalData: 'I am global data'
})
```


操作：

1、App.prototype.getCurrentPage()

getCurrentPage() 函数用于获取当前页面的实例。

2、getApp()

我们提供了全局的 getApp() 函数，可以获取到小程序实例。

```
// other.js
var appInstance = getApp()
console.log(appInstance.globalData) // I am global data
```

注意：

1. App() 必须在 app.js 中注册，且不能注册多个。
2. 不要在定义于 App() 内的函数中调用 getApp()，使用 this 就可以拿到 app 实例。
3. 不要在 onLaunch 的时候调用 getCurrentPage()，此时 page 还没有生成。
4. 通过 getApp() 获取实例之后，不要私自调用生命周期函数。

2、app.json： 设置界面组成、公共窗口展现、底部 tabbar 样式、超时时间等

```
{
  "pages": [
    "pages/index/index",
    "pages/logs/index"
  ],
```

```
"window": {
  "navigationBarTitleText": "Demo"
},
"tabBar": {
  "list": [{
    "pagePath": "pages/index/index",
    "text": "首页"
  }, {
    "pagePath": "pages/logs/logs",
    "text": "日志"
  }]
},
"networkTimeout": {
  "request": 10000,
  "downloadFile": 10000
},
"debug": true
}
```

3、 app.wxss：配置公共样式

定义在 app.wxss 中的样式为全局样式，作用于每一个页面~

```
.container {
  font-family: Helvetica Neue, Helvetica, Arial, sans-serif;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
```

```
    box-sizing: border-box;
}
```

页面

一个框架页面由四个文件组成，分别是：

文件类型	必填	作用
js	是	页面逻辑
wxml	是	页面结构
wxss	否	页面样式表
json	否	页面配置

对于小程序很多人有一个错觉，那就是小程序是用 **HTML5**，这个真说错了，看看页面的**架构**， 小程序是微信全新定义的规范，是基于 **XML+JS** 的，不支持也不兼容 **HTML**，兼容受限的部分 **CSS** 写法。

上图的小程序的 page 目录，后缀名分别是 wxml、wxss 和 js，不过微信对 wxml 的全称定义也不是 weixin xml，而是 **WeiXin Markup Language**，同时 wxss 也是 **WeiXin Style Sheets**，也说明了小程序开发将是脱离传统**前端开发**的一种形态，下文介绍四部分的详细应用~

index.json :

界面的配置，只能设置 window 相关的配置项，以决定本页面的窗口表现，包括 navigationBar 的标题、背景、颜色及是否允许下拉等

```
{
  "navigationBarBackgroundColor": "#ffffff",
```

```
"navigationBarTextStyle": "black",  
"navigationBarTitleText": "微信接口功能演示",  
"backgroundColor": "#eeeeee",  
"backgroundTextStyle": "light"  
}
```

注意：

为了方便开发者减少配置项，规定描述页面的这四个文件必须具有相同的路径与文件名。

简单示例

我的水滴



walid

我的账户 >

邀请好友 (一起赚钱) >

我的业绩 >

用户协议 >

当前版本 V 1.0.0 (内测版)

客服电话 400-686-1179

服务时间 10:00~22:00

官方邮箱 dida@shuidihuzhu.com

一、 编写 app.js

```
//app.js
App( {
  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,
  },
})
```

二、配置 app.json

增加个人中心界面

```
{
  "pages": [
    "pages/mine/index"
  ],
  "window": {
    "backgroundTextStyle": "light",
    "backgroundColor": "#ffffff",
    "navigationBarBackgroundColor": "#0092ff",
    "navigationBarTitleText": "滴答宝",
    "navigationBarTextStyle": "white"
  }
}
```

三、编写个人中心界面

目录结构如下：

```
▼ mine
  JS index.js
  JN index.json
  <> index.wxml
  { } index.wxss
```

1、index.wxml

```
<!--index.wxml-->
<toast hidden="{{toastHidden}}" duration="1000" bindchange="toastChange">{{toastContent}}</toast>
<view class="container">
  <view bindtap="headTap" class="userinfo">
    <image class="userinfo-avatar" src="{{userInfo.avatarUrl}}" background-size="cover"></image>
    <text class="userinfo-nickname">{{userInfo.nickName}}</text>
  </view>
  <block wx:for-index="index" wx:for="{{userListInfo}}">
    <view data-index="{{index}}" bindtap="cellItemClick" class="weui_cell" style="margin-top:{{item.margin}}px;">
      <view wx:if="false" class="weui_cell_hd">
        <image src="{{item.icon}}"></image>
      </view>
      <text class="tv_cell_left"> {{item.leftItem}} </text>
      <text wx:if="{{item.rightItem}} == null" class="tv_cell_right">{{item.rightItem}}</text>
      <view wx:if="{{item.showArrow}}" class="iv-arrow"></view>
    </view>
  </block>
  <view class="v_below">
    <text class="tv_server_time">客服电话 400-686-1179</text>
    <text class="tv_server_time">服务时间 10:00~22:00</text>
    <text class="tv_server_time">官方邮箱 dida@shuidihuzhu.com</text>
  </view>
</view>
```


2、index.wxss

```
.container {  
  background-color: #F2f2f2;  
}  
  
.userinfo {  
  display: flex;  
  flex-direction: column;  
  padding: 50rpx 0;  
  align-items: center;  
  background: #0092ff;  
}  
  
.userinfo-avatar {  
  width: 128rpx;  
  height: 128rpx;  
  margin: 20rpx;  
  border-radius: 50%;  
}  
  
.userinfo-nickname {  
  color: #333;  
}  
  
.weui_cell {  
  position: relative;  
  display: flex;  
  padding: 15px;  
  -webkit-box-align: center;  
  -ms-flex-align: center;  
  align-items: center;  
  border-bottom: 1px solid #dadada;
```

```
    background-color: white;
}

.weui_cell_hd image {
    width: 100%;
    height: 20px;
    vertical-align: -2px;
}

.tv_cell_left {
    position: absolute;
    font-size: 30rpx;
    color: #333;
    left: 20rpx;
}

.tv_cell_right {
    position: absolute;
    font-size: 28rpx;
    color: #999999;
    right: 50rpx;
}

.iv-arrow {
    position: absolute;
    right: 30rpx;
    border-top: 2rpx solid #c7c7c7;
    border-right: 2rpx solid #c7c7c7;
    width: 16rpx;
    height: 16rpx;
    transform: rotate(45deg);
    -webkit-transform: rotate(45deg);
}
```

```
.v_below {
  display: flex;
  flex-direction: column;
  align-items: center;
}

.tv_server_time {
  font-size: 28rpx;
  color: #999999;
  margin-top: 20rpx;
}
```

3、index.js

```
var app = getApp()
```

```
Page( {
  data: {
    userInfo: {},
    toastHidden: true,
    toastContent: '',
    userListInfo: [ {
      leftItem: '我的账户',
      showArrow: true,
    }, {
      leftItem: '邀请好友 （一起赚钱）',
      margin: '20rpx',
      showArrow: true,
```

```
    }, {
        leftItem: '我的业绩',
        showArrow: true,
    }, {
        leftItem: '用户协议',
        margin: '20rpx',
        showArrow: true,
    }, {
        leftItem: '当前版本',

        rightItem: 'V 1.0.0 (内测版)',
        margin: '20rpx',
        showArrow: false,
    }
]

},

onLoad: function() {
    var that = this

    //调用应用实例的方法获取全局数据
    app.getUserInfo( function( userInfo ) {

        //更新数据
        that.setData( {
            userInfo: userInfo
        })
    })
}

},

toastChange: function() {
```

```
        this.setData( {
            toastHidden: true
        })
    },

    headTap( e ) {
        //更新数据
        this.setData( {
            toastHidden: false,
            toastContent: "点击头部"
        })
    },

    cellItemClick( e ) {
        var index = e.currentTarget.dataset.index
        console.log( "index = " + index )

        //更新数据
        this.setData( {
            toastHidden: false,
            toastContent: "点击" + index
        })
    },
})
```

整体代码十分简单，这里只是想让您看到基本的效果演示，如果需要更深层次的使用，大家还请到官方文档阅读，相信您几个小时就可以搞定，简

单的集成演示就介绍到这里~

技术缺陷

一定会有人发出疑问，微信小程序就那么优秀吗？没有问题、没有坑吗？

答：肯定是有的，在这里草民表达一下自己对程序框架的看点，代码世界的框架只有两种，第一种是天天被吐槽的，第二种是没人用的，希望大家还是要以发现美的眼光去看待一个框架，同时还要用挑剔的眼光去审视一个框架，采用发现问题，提出问题，解决问题的方式进行优化。

下面我介绍点在微信小程序开发时一些不爽的地方，其他问题都是可以解决的，草民只是提几个代码层级的问题，这也是我最注重的：

1、操作反馈组件

小程序提供了四种操作反馈组件，`action-sheet`、`modal`、`toast`、`loading`，想法很好，而且多数框架都有，可草民不理解的就是，您不能封装成 `module` 来使用吗？

看下官方使用方式：

示例代码：

```
<view class="body-view">
  <toast hidden="{{toast1Hidden}}" bindchange="toast1Change">
    默认
  </toast>
  <button type="default" bindtap="toast1Tap">点击弹出默认toast</button>
</view>
<view class="body-view">
  <toast hidden="{{toast2Hidden}}" duration="3000" bindchange="toast2Change">
    设置duration
  </toast>
  <button type="default" bindtap="toast2Tap">点击弹出设置duration的toast</button>
</view>
```

```
var toastNum = 2
var pageData = {}
pageData.data = {}
for(var i = 0; i <= toastNum; ++i) {
  pageData.data['toast'+i+'Hidden'] = true
  ;(function (index) {
    pageData['toast'+index+'Change'] = function(e) {
      var obj = {}
      obj['toast'+index+'Hidden'] = true
      this.setData(obj)
    }
    pageData['toast'+index+'Tap'] = function(e) {
      var obj = {}
      obj['toast'+index+'Hidden'] = false
      this.setData(obj)
    }
  })(i)
}
Page(pageData)
```

我希望的方式是这样的：

```
var toast = require('@wx-module/model')

modal.toast({
  'message': 'toast 消息内容',
  'duration': 1000
})
```

不知道大家的想法是怎样的，反正草民是非常遵循傻瓜式编程的~

2、template 模板封装

和大家表明一下，小程序的 **template** 模板只支持 **wxml** 的引入，并不支持具体的业务逻辑与 **css** 样式，这也是本人接受不了的现实，我想封装一个组件，组件中没有业务逻辑，我管他叫什么？组件化开发是这样定义的吗？希望腾讯下一版本迭代是将此问题解决。

3、不支持配置别名，只能采用相对路径的方式进行 **require**

看一下 **require** 的方式：

```
var router = require('../../config/router.js')

草民想冒昧问一句，这个../../您看着爽吗，和下面这个比如何？

var router = require('root/config/router.js')
```

4、适配方案 **rpx**（responsive pixel）

微信小程序新定义了一个尺寸单位，可以适配不同分辨率的屏幕，它规定屏幕宽为 **750rpx**，如在 **iPhone6** 上，屏幕宽度为 **375px**，共有 **750** 个物理

像素，则 $750\text{rpx} = 375\text{px} = 750$ 物理像素， $1\text{rpx} = 0.5\text{px} = 1$ 物理像素。

设备	rpx换算px (屏幕宽度/750)	px换算rpx (750/屏幕宽度)
iPhone5	$1\text{rpx} = 0.42\text{px}$	$1\text{px} = 2.34\text{rpx}$
iPhone6	$1\text{rpx} = 0.5\text{px}$	$1\text{px} = 2\text{rpx}$
iPhone6s	$1\text{rpx} = 0.552\text{px}$	$1\text{px} = 1.81\text{rpx}$

项目中我大多数都采用 rpx 进行适配，但是有时 1rpx 有时显示不出来，而且当我设置 750rpx 时，在 android 手机显示不满一屏，就是说在 android 屏幕上并没有按照模板 750rpx 进行适配或者说适配方案存在问题，也希望开发者在使用时注意。

运营问题：

送大家一个彩蛋吧，在这里提醒一下大家运营小程序时可能出现的问题，希望大家在运营时避免以下几点，防止犯一些不必要的错误导致影响运营效果与项目进度~

提交「小程序」时需要注意哪些地方，才不会被拒？

1、小程序的名称必须和所提供的服务有所关联，而且必须是两个词以上的组合，不能使用广泛使用不具备识别性的词语来命名，比如日历、电话。

- 2、提交小程序审查时，头像 LOGO 要尽量使用清晰度高的图片，这个清晰度是指可以看清、分别、图片中包含的元素，否则微信不予通过。
- 3、微信希望，用户打开小程序的时候直接就能使用到相关的功能，因此这份文档提到，用户在小程序的页面就能使用到相关的服务，不能隐藏，也不能进行多次跳转。有特别规定，「小程序首页必须能够直达或者经过 2 次点击到达」相关的功能。
- 4、小程序的功能不能过于简单，比如说只有一个页面，或只提供一个按钮。
- 5、小程序不能展示、推荐第三方小程序，不能做小程序导航、互推、小程序排行榜等。也不能通过小程序来搜索其它小程序。
- 6、用户使用小程序时，不能以关注或使用其它小程序作为条件。
- 7、禁止诱导分享，这个不用再说，微信对此一直严厉打压。
- 8、小程序不能用作营销活动、广告用途，不能存在类似算命、星座运势之类的测试类内容，不能包含赌博竞猜抽奖等内容。
- 9、小程序里面的图片，不能包含广告、网址。

运营「小程序」的时候，需要注意哪些地方，才不会下架？

- 1、微信不允许批量注册、重复提交大量类似的小程序。
- 2、小程序的添加，不能设置付费，必须是免费添加。
- 3、如果你的微信小程序含有支付功能，需要在简介当中明确指示。

4、如果你希望通过小程序创业，需要注意：如果小程序所属的公司被收购或合并，你从微信和微信小程序接收的小程序里继续使用。

5、如果小程序使用到地理位置定位功能，在采集、 获取用户地理位置时必须得到用户的同意。

6、小程序后台服务，仅限于集中目的：**VoIP**、音频播放、地理位置、完成任务和本地提醒。

7、禁止多媒体比如音频、视频的自动播放。

结语

用了小半天的时间，终于把小程序分享完毕了，希望能够对大家有所帮助，谢谢（-_-）