

初探微信小程序：大致流程，实现机制及开发前须知

分享者：treadpit 原文地址

一、小程序是什么？

- 一种介于原生 app、和 web app 的 hybrid, 比 web app 的开发成本还低;
- 通过微信进行加载;
- 相对原生 app 来说，更加轻量、更新实时、跨平台;
- 相对 web app 来说，资源离线，体验更流畅;
- 低频、无强粘性;
- 可以使用微信的支付功能

二、接入小程序的大致流程：

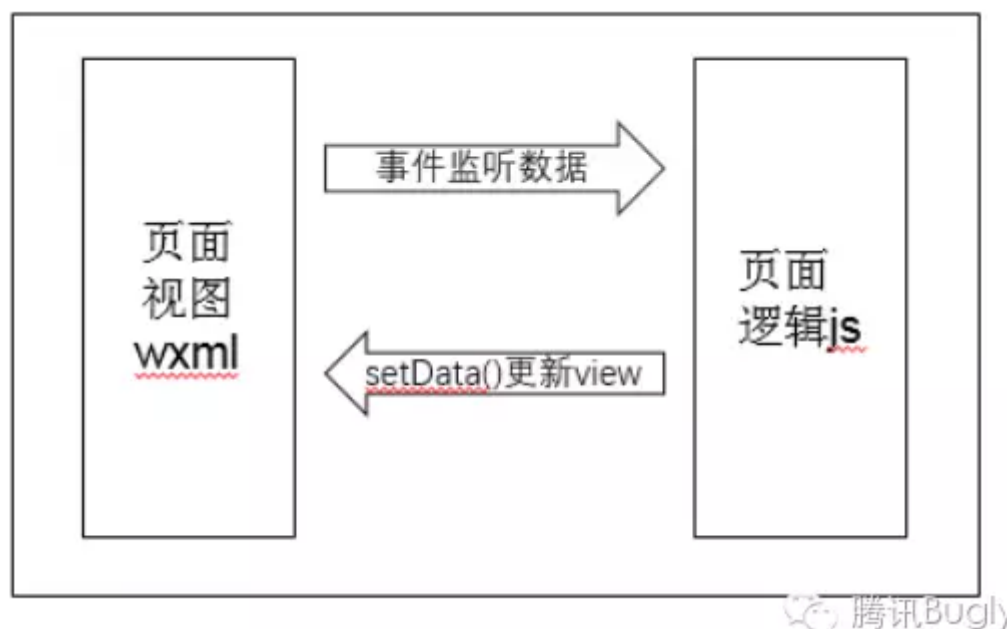
- 注册小程序：尚未对个人开放，需要对公打款或完成微信认证才有微信支付功能;
- 资料完善：包括小程序命名、描述、服务范围等;
- 功能开发;
- 提交审核：需要保证功能的完整性。

[开发者工具](#) [开发文档](#)

三、实现机制

基于微信提供的一套应用框架。微信通过封装微信客户端提供的**文件系统**、**网络通信**、**任务管理**、**数据安全**等基础功能，对上层提供了一套完整的 **Javascript Api**，开发者能够非常方便的调用微信客户端提供的各种基础功能。

视图层描述语言 .wxml 和 .wxs，以及基于 JavaScript 的逻辑层框架，并在视图层与逻辑层之间通过 **单向数据** 绑定进行数据传输



四、开发前须知

- AppStore 问题;
- 发布的项目包体积 < 1M, 只适合轻量级;
- 由于现在官方的限制，只能 7 天内能够给使用过该小程序的用户进行消息发送，所以关于消息发布还需要更多的斟酌;

- 开发基于微信框架，部分功能受限，不能操作 DOM，没有 script 标签，要引入则需加入到项目；
- 没有与浏览器 BOM 相关的 API；
- 没有 cookie 🤖；
- 用 **storage** 替代了 H5 中的 localStorage、sessionStorage、storage 对每个小程序的大小是 **10M**，支持 **同步** 和 **异步** 🤔。
- 同时只能存在 **5** 个 url 请求；
- 小程序页面只能同时打开 **5** 个，如果交互流程较长难以支持；
- 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；
- 其他注意事项 [设计文档](#) [运营规范](#)

注意：cookie 问题，在请求发送时，可以动态设置 Header 发送报文的 cookie，但是 **cookie 本身不能在客户端进行读写**。

```
wx.request({  
  
  header: {cookie: "abcd1234"},  
  
  url: "/getinfo",  
  
  data: {},  
  
  ...  
  
})
```

五、路由

1. pages 里面的第一个元素即为首页;
2. 每个页面需要手动在 app.json 中进行注册，否则不能访问;
3. 路由跳转： 组件跳转(navigator) / API 跳转;
4. 只能同时打开 5 个页面，否则 wx.navigateTo 不能正常打开新页面，避免多层级的交互方式，或者使用 wx.redirectTo。

// <navigator/>组件内部不能再嵌套<navigator/>组件。只能是单层存在

```
<navigator url="search/search">
```

```
  <view class="serach_view_show" bindtap="bindtap"> 搜索</view>
```

```
</navigator>
```

// API

```
wx.navigateTo({ // 保留当前页

  url: "",

  ...

})
```

```
wx.redirectTo({ // 关闭当前页

  url: "",

  ...

})
```

六、Tab 页

1. 由 app.json 定义；
2. 最多 **5** 个；
3. 每个页面的 .json 文件可以覆盖定义导航栏。

七、页面结构

由同路径下同名的四个不同后缀文件的组成：

- .js 文件是脚本文件
- .json 文件是配置文件
- .wxss 是样式表文件
- .xml 文件是页面结构文件



app.json 必须，微信框架配置文件入口，整个小程序的全局配置。包括页面注册，网络设置，以及小程序的 window 背景色，配置导航条样式，配置默认标题。

app.js 必须，可以什么都不需要写，后期再监听并处理小程序的生命周期函数、声明全局变量。

.xml 文件

// wxml 语法

```
<block wx:for="{{shareItems}}" wx:for-index="idx">
```

```
<view id="{{idx}}" class="flex share-item box box-tb">

  <view class="flex box box-lr article-info">

    <view class="flex title">{{item.articleInfo.title}}</view>

  </view>

  <view class="user-info box box-lr">

    <image src="{{item.articleInfo.isApproved ? icon.approved : icon.approve}}" class="icon approve-icon"> </image>

  </view>

</view>

</block>
```

1. .wxml 文件通过相同的名称，将页面与逻辑js、样式、配置进行关联匹配
2. .wxml [组件语法](#)
3. .wxml 提供两种文件引用方式 [import\(有作用域\)](#)和 [include](#)
4. 事件绑定方式

```
// test.wxml
```

```
<view class="flex title" bindtap="getInfo">{{userName}}</view>
```

```
// test.js
```

```
page({  
  
  data: {},  
  
  getInfo: function() {  
  
    this.setData({  
  
      userName: "Tom"  
  
    })  
  
  }  
  
})
```

.wxss 文件

```
/** app.wxss */
```



```
@import "common.wxss";
```

```
view {  
  
  padding:15px;  
  
}
```

1. 可通过 `@import` (后跟相对路径)样式表;
2. **尺寸单位** : 有两种单位 : (1)、引入 `rpx`(根据屏幕宽度进行自适应)的概念 ; (2)、`rem` ;
3. 支持内联
4. **选择器** : 不再支持媒体查询 , 支持 `.demo`, `#demo`, `view`, `view input`, `view::after`, `view::before`;
5. 增加了 app 的 flex 布局;
6. **weui 官方样式库** 🙄

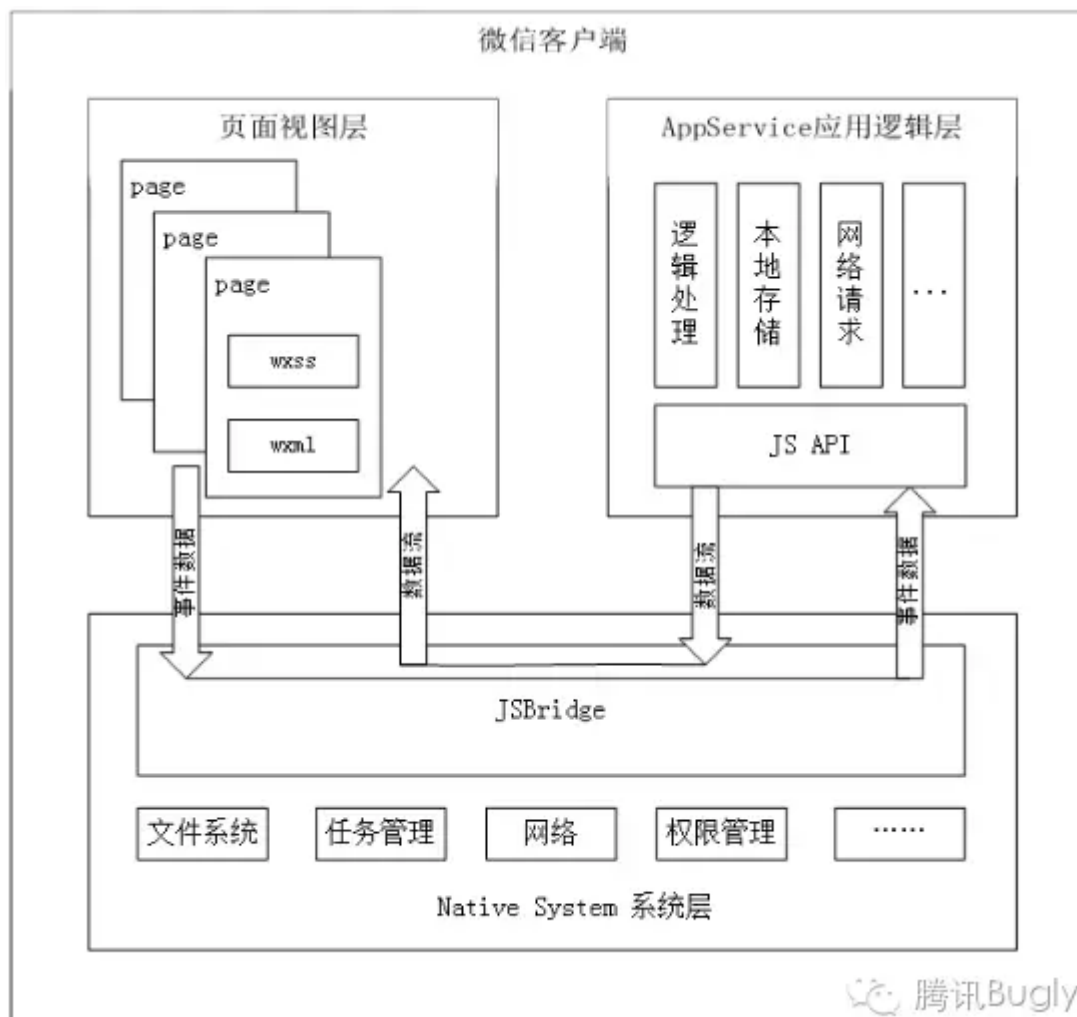
规定屏幕宽为 750rpx, 如 iPhone6, $1\text{rpx} = 1 \text{ 物理像素} = 0.5\text{px}$;

规定屏幕宽度为 20rem, $1\text{rem} = (750/20)\text{rpx}$;

.js 文件

1. 模块运行 (类似 node , 框架自动添加外层 define) ;

2. 形式上支持 CommonJs , 通过 require 加载;
3. data 应约定为只读 , 不能直接修改 data 值 , 否则 **容易造成 data 中的数据与 view 不一致**;
4. 更新 View 需使用 `setData()` , 与 data 中的数据进行 Diff 比较 , 不同才会更新。
 - `setData()` 单次设置的数据 < **1M** , 要避免一次设置过多的数据;
 - 支持 ES6 中的 ... 操作符展开模块数据。



```
// js 引入
```

```
// a.js
```

```
function sayHi() {
```

```
    console.log("hello wxAPP!");  
}
```

```
module.exports = {  
    say: sayHi  
}
```

```
// b.js
```

```
var say = require("a.js");
```

```
// data 操作
```

```
page({  
    data: {  
        name: 'one'  
    },  
    handleData() {
```

```
// 错误操作方式，虽不会报错，但不会更新 view

this.data.name = 'two';


// 正确操作方式

this.setData({

  name: 'tow'

})

}

})
```

.json 文件

- 各个页面的配置性文件

八、事件绑定

事件类型	描述
touchstart	手指触摸动作开始
touchmove	手指触摸后移动
touchcancel	手指触摸动作被打断，如来电提醒，弹窗
tap	手指触摸后马上离开
longtap	手指触摸后，超过 350ms 再离开

bind 或 catch 开头，然后跟上事件的类型，如 bindtap bind 事件绑定不会阻止冒泡事件向上冒泡，catch 事件绑定可以阻止冒泡事件向上冒泡。

```
<view bindtap="clickMe">绑定事件</view>
```

八、杂项

微信支付

- 申请微信支付
- 设置密钥和下载证书

- 配置 Https 服务器
- 小程序与 JSSDK 微信支付比较

data 操作

```
Page({  
  
  data: {  
  
    user: {  
  
      name: {  
  
        nickName: 'Tom'  
  
      }  
  
    }  
  
  },  
  
  changeData() {  
  
    // data 设置，key 可以非常灵活，以数据路径的形式给出，如 array[2].message，a.b.c.d  
  
    this.setData({  
  
      'user.name.nickName': '汤姆'  
  
    })  
  
  }  
})
```

```
// 可以不预先在 this.data 中定义

this.setData({

  'sex': 'male'

})

}

})
```

// 标签自定义 data 取值

// index.wxml

```
<view data-name="tear" data-type="someType" bindlongtap="getData">绑定事件</view>
```

// index.js


```
Page({  
  getData(e) {  
    const _data = e.currentTarget.dataset; // => {name: 'tear', type: 'someType'}  
  }  
})
```

十、坑

- 编辑器问题；
- 调试工具和手机表现不一致问题(如 弹框标题);
- API 在调试工具和真机上返回值不一样（如 wx.showModal()）；
- 模拟器对 css 的支持比较完整，真机只支持部分，但是文档没有
- 列出来;
- 如果 api 是 post 请求,参数是 key-value 形式,那么直接设置 Object 是不行的,必须手动拼;
- view 标签不识别'/n'换行符,text 组件可以;
- toast 有 success 和 loading 状态,竟然没有失败状态，宽度是写死的,字一多就丑得要死;
- navigateTo 无法封装到另一个 js 方法中,只能直接写,而且失败还不走失败的回调;

- 元素不能获取自定义 data 的完整对象值,只能传基本数据类型;
- 开发工具更新后,原先 height:auto 失效,必须指定 image 的高度为具体数值,不然高度为 0。
- text 可以嵌套 text,但不可以嵌套 view
- button 的边框是用:after 方式实现的,用户如果在 button 上定义边框会出现两条线
- 分隔线高度使用 rpx 问题