

徐德明：微信小程序浅析

目录

微信小程序是什么？
微信小程序与公众号有什么区别？
微信小程序的语法？

项目结构
页面管理
基础组件
丰富的 API
项目运行过程
开发者工具（IDE）

DEMO

目前遇到的问题
微信小程序是什么？

微信小程序是一种不需要下载安装即可使用的应用，它实现了应用「触手可及」的梦想，用户扫一扫或搜一下即可打开应用。

目前是内测阶段，而且采用的是邀请制的方式，据说只有 200 个。

<https://mp.weixin.qq.com/debug/wxadoc/dev/?t=1477656487237>

微信小程序与公众号有什么区别？

在没有网络连接的情况下，微信公众号的功能无法使用，但微信小程序而言，只要小程序本身无需联网的工作，它能够在离线的作用下发挥作用。

服务号与在微信中加载一般网页相比，小程序的加载十分快捷，会给用户一种使用流畅的感觉。而公众号里无论是访问图文消息，还是打开第三方开发的网页，相比起小程序而言，都会慢许多。

微信为小程序提供了更强的绘图能力、丰富的界面控件和更全的操作反馈，这一切使得微信小程序会拥有更好的显示效果和交互能力。

小程序不会和公众号一同挤在会话列表中，除了通过一个会话可以直接打开小程序，也能到小程序自己的存放列表搜索和打开小程序。预计小程序的列表入口会放在 “发现” \ “游戏” 的下面。

微信小程序提供了更强的网络连接能力，小程序能够更加自由的连接网络，也能更加安全的使用网络。能够便捷地访问智能设备链接，未来在微信中就能直接使用和操作其他智能硬件。

微信小程序提供了成熟的媒体组件，微信小程序的开发者可以更容易地开发视频/音频等多媒体应用，用户使用起来也会更加的流畅。

微信小程序不能够向用户推送图文，只能发送模板消息，从而不会无事打扰用户，也失去了通过图文消息激活用户和提高用户粘度的能力。

微信小程序的语法？

微信提供了丰富的框架组建和 **API** 接口供开发者调用，具体包含：界面、视图、内容、按钮、导航、多媒体、位置、数据等等。这些组件和接口能让建立在微信上的小程序在运行能力和流畅度上保持和 **Native APP** 一样的体验。

项目结构

最关键也是必不可少的，是 **app.js**、**app.json**、**app.wxss** 这三个。 **img**

app.js 小程序运行主要逻辑及入口，里面使用 **App()**函数来注册一个小程序，普通页面的 **js** 文件中可以通过 **getApp()**函数拿到 **App()**函数所拥有的参数，并调用其中的数据。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。小程序里面的 **JS** 文件都是模块化的，模块只有通过 **module.exports** 或者 **exports** 才能对外暴露接口。

```
App({
  onLaunch: function() {
    //当小程序初始化完成时，会触发 onLaunch（全局只触发一次）
  },
  onShow: function() {
    //当小程序启动，或从后台进入前台显示，会触发 onShow
  },
  onHide: function() {
    //当小程序从前台进入后台，会触发 onHide
  },
 .globalData: 'I am global data'
})
```

app.json 是小程序的全局配置文件。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。

```
{
  "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
  }
}
```

app.wxss 是一个公共的样式文件，整个项目的每个页面都可以调用，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，就如一个全局的 css 文件。小程序引入了一个 rpx 的尺寸单位（会内部转成 rem），也可以用普通的单位，如 px、em、rem、百分比等；

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

```
.container {
  height: 100%;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: space-between;
  padding: 20rpx 0;
```

```
    box-sizing: border-box;
    padding: 10rpx;
}
```

页面管理

框架管理了整个小程序的页面路由，可以做到页面间的无缝切换，并给以页面完整的生命周期。框架采用的是双向数据绑定 **MVVM** 的模式。

每个页面由 4 个文件组成，分别是页面逻辑文件 **JS**、页面结构文件 **WXML**、页面样式文件 **WXSS** 和页面配置文件 **JSON**（这里的配置会覆盖全局的配置，即 **app.json**）。微信小程序会读取这些文件，并生成小程序实例。

小程序拥有全局的 **App** 和 **Page** 方法，用于进行程序和页面的注册，并且其中定义的方法可以被相互调用。

```
Page({
  data: {
    text: "This is page data."
  },
  onLoad: function(options) {
    // Do some initialize when page load.
  },
  onReady: function() {
    // Do something when page ready.
  },
  onShow: function() {
    // Do something when page show.
  },
  onHide: function() {
    // Do something when page hide.
  },
  onUnload: function() {
    // Do something when page close.
  },
  // Event handler.
  viewTap: function() {
    this.setData({
      text: 'Set some data for updating view.'
    })
  }
})
```

```
}  
});  
WXML
```

```
<!--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>
```

基础组件

框架提供了一套基础的组件，这些组件自带微信风格的样式以及特殊的逻辑，开发者可以通过组合基础组件，创建出强大的微信小程序。具体基础组件有 **view**、**image**、**text**、**input**、**audio**、**canvas** 等等，每个组件都有非常丰富的属性及事件绑定；

```
<video id="myVideo" src="{{src}}" binderror="videoErrorCallback"  
  bindplay="bindplayCallback" autoplay controls></video>
```

丰富的 API

提供了丰富的微信原生 API，可以方便的调起微信提供的能力，如请求接口、登录，本地存储，上传下载等；

```
wx.request({  
  url: 'test.php', //仅为示例，并非真实的接口地址  
  data: {  
    x: "",  
    y: ""  
  },  
  header: {  
    'Content-Type': 'application/json'
```

```
},
success: function(res) {
  console.log(res.data)
}
})
```

项目运行过程:

第一步:加载项目根目录下的 `app.js`、`app.json`、`app.wxss` 文件, 同时会执行 `app.js` 文件, 并触发其中的 `onLaunch` 和 `onShow` 函数;

第二步:加载 `app.json` 中 `pages` 数组中配置的第一个页面, 作为项目的欢迎页, 同时会执行对应页面 `js` 文件, 并触发 `onLoad / onReady` 和 `onShow` 函数;

往后:页面可以通过事件与 `js` 文件交互, 比如 在标签元素上绑定点击事件, 并且指向 `js` 文件中的一个函数, 就能用 `js` 中的逻辑去处理这个事件了

开发者工具 (IDE)

为了帮助开发者简单和高效地开发微信小程序, 我们推出了全新的开发者工具, 集成了开发调试、代码编辑及程序发布等功能。IDE 采用的是 `node-webkit` 加 `react`

编辑功能

开发者可以在这里编写代码, 并且有适当的代码提示功能。(建议开发者用专业的 IDE 开发代码, 然后可以利用此工具预览代码);

调试功能

这里可以预览开发的小程序, 并且提供了丰富的调试工具, 分为 6 大功能模块: `Wxml`、`Console`、`Sources`、`Network`、`Appdata`、`Storage`;

在 IDE 上面一栏可以选择手机模式和网络模式, 用于模拟各种终端及网络环境下的小程序运行情况;

项目功能

在这一栏可以设置小程序项目的本地路径, 可以上传小程序, 生成预览的二维码和一些可选项等等; 注: 上传及预览需要 `appid`;

下载地址

<https://mp.weixin.qq.com/debug/wxadoc/dev/devtools/download.html?t=1477656486010>

demo

由于目前没有 `appid`, 只能在 IDE 中预览小程序

目前遇到的问题

由于目前还只是内测阶段 **beta** 版本，很多功能及 **bug** 也正在 **fix** 中，微信团队基本按照 2 周一次的节奏发布版本更新;

小程序技术规范刚刚起步，复杂的需求还缺乏实施方式;

由于小程序不是基于 **webview**，而是运行在 **JS core** 中，所以没有 **window** 和 **document** 对象等，而且是 **MVVM** 双向数据绑定的模式，无法进行 **DOM** 的操作，所有的数据操作必须基于组件上的事件绑定，利用 **setData** 方式改变初始状态，一些浏览器插件也无法使用（因为插件基本都基于 **window** 和 **doucument** 对象），所有开发模式上要有一定的改变;

图片没法根据宽度自适应高度，必须写死高度的值，这给开发造成一定的困扰，特别是宽度为百分比的时候;

微信提供了调用接口的方法，但是必须是 **https** 协议;

一些功能必须依赖于 **appid**，比如登录，**socket**，支付等等;

开发动画的时候，初始样式必须写 **style** 属性，写 **CSS** 没用;

etc...