

微信小程序开发

在开发方面，微信为开发者提供了较为完善的开发文档，提供的接口和框架也非常的丰富。还提供了跨平台的开发工具，使不同平台都可以开发小程序。

组件参考：

- 视图容器：View、Scroll-View、Swiper
- 基础内容：Icon、Text、Progress
- 表单组件：Button、CheckBox、form、Input等
- 操作反馈：Action-Sheet、Modal、Loading
- 导航：Navigator
- 媒体组建：Audio、Image、Video
- 地图：Map
- 画布：Canvas

API 参考

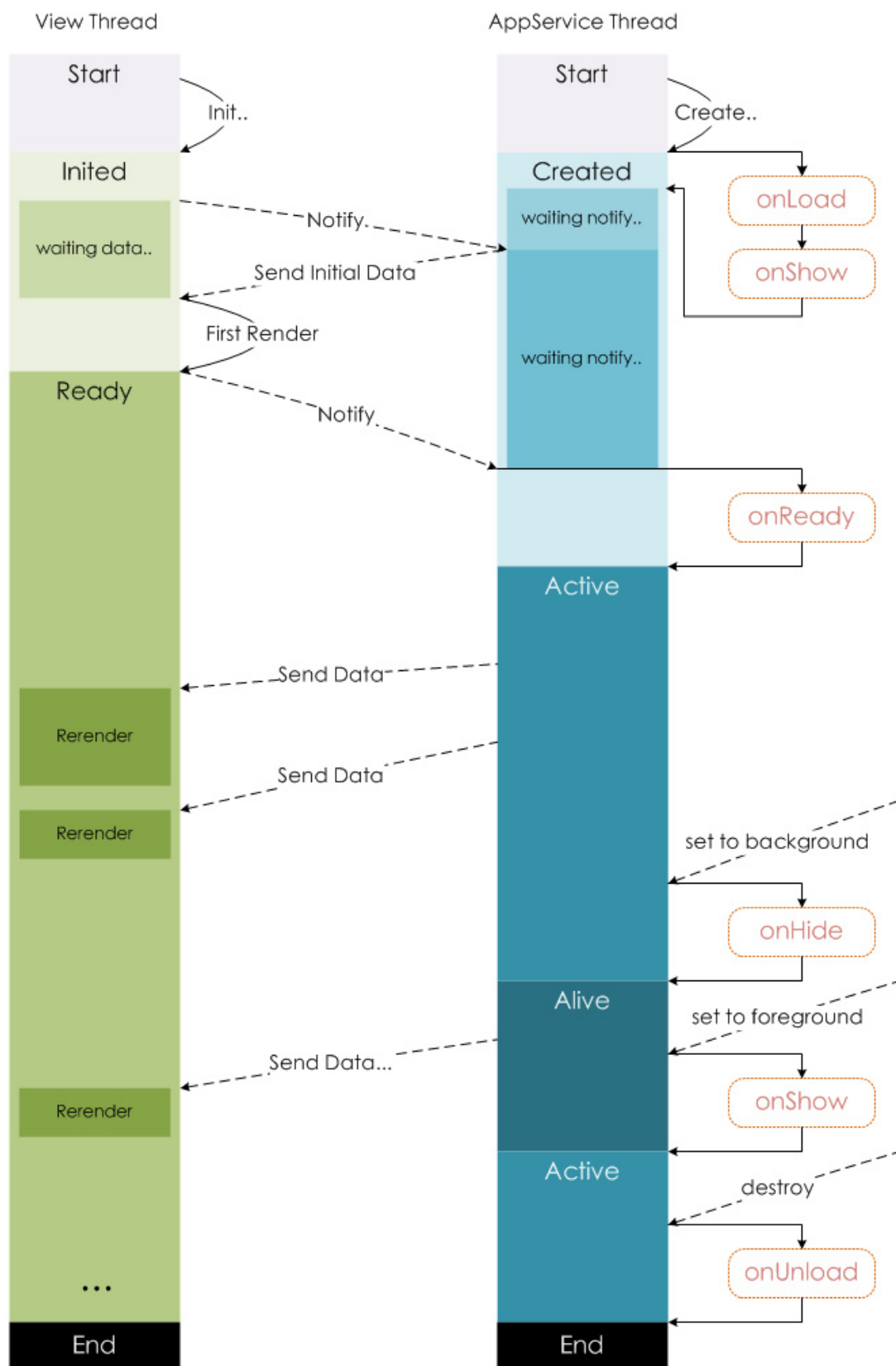
- 网络：常规请求、上传下载、WebSocket
- 媒体：图片、录音、音频/音乐播放控制、文件操作、视频播放
- 数据：数据缓存能力
- 位置：获取位置、查看位置
- 设备：网络状态、系统信息、重力感应、罗盘
- 界面：设置导航条、导航、动画、绘图、其它
- 开放接口：登录（签名加密）、用户信息、微信支付、模板消息

小程序的开发偏向组件化、结构化。小程序加载包到本地运行，与 Web 模式有很大区别，不能操作DOM，通过 API 与服务器交互，更像 C/S 架构的开发。适合逻辑简单的工具型应用，很强的内容封闭性。

微信设计了一套框架：**MINA**，框架提供了自己的视图层描述语言 WXML 和 WXSS，以及基于 JavaScript 的逻辑层框架，并在视图层与逻辑层间提供了数据传输和事件系统，可以让开发者可以方便的聚焦于数据与逻辑上。

框架的核心是一个响应的数据绑定系统。整个系统分为两块视图层（View）和逻辑层（App Service） 框架可以让数据与视图非常简单地保持同步。当做数据修改的时候，只需要在逻辑层修改数据，视图层就会做相应的更新。

MINA 生命周期



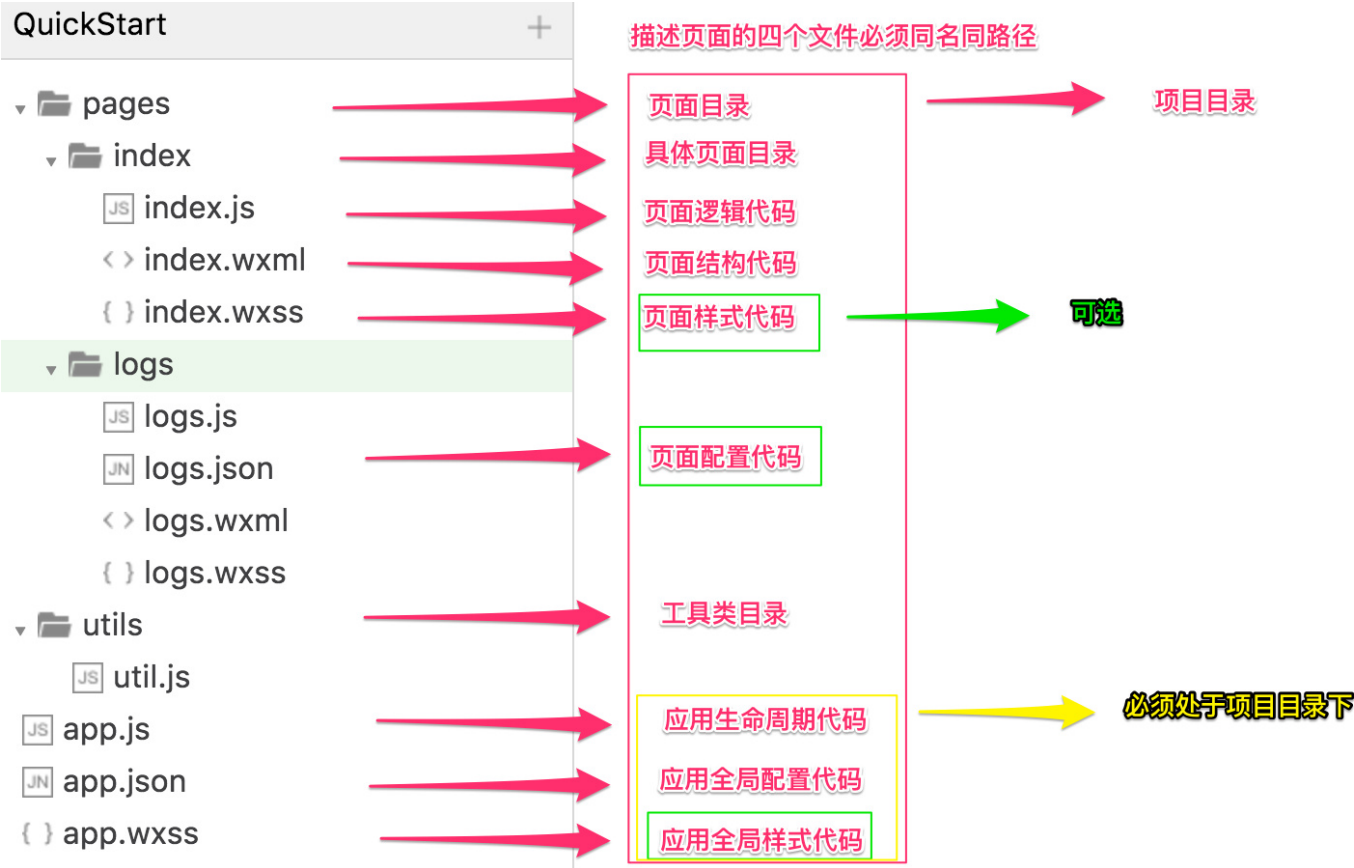
目录结构

项目根目录全局文件

- `app.js`: 小程序逻辑 (生命周期、全局变量、全局方法)
- `app.json`: 小程序全局设置
- `app.wxss`: 小程序公共样式表

页面目录文件

- .js: 页面逻辑
- .wxml: 页面机构
- .wxss: 页面样式
- .json: 页面配置



全局配置文件(app.json)

属性	类型	必填	描述
pages	String Array	是	设置页面路径
window	Object	否	设置默认页面的窗口表现
tabBar	Object	否	设置底部 tab 的表现
networkTimeout	Object	否	设置网络超时时间
debug	Boolean	否	设置是否开启 debug 模式

界面搭建

1. 基本逻辑

WXML和WXSS两种文件是小程序界面元素声明及样式描述文件。

WXML最大的特点是以视图（View）的方式串联界面元素，并通过程序逻辑（AppService）将信息更新实时传递至视图层。

View类似于HTML中的div元素，在构建的时候，View可以被多级嵌套，View内可以放置任意视觉元素。

需要注意的是，元素一旦超出屏幕之外，用户就无法看到了，这是与HTML较大的不同。小程序有专门用于滚动的视图。如果希望界面是一个可以自由滚动的界面（例如列表等），可以使用scroll-view视图，在WXSS中将其大小调整为整个屏幕，并设置scroll-y（上下滚动）或scroll-x（左右滚动）为true。

注意：小程序中不能直接使用DOM控制WXML元素。如果需要数据进行更新，就要使用WXML提供的数据绑定及元素渲染方法，还有一点，小程序的栅格排版系统使用的是 [flex](#) 布局。

2. 绑定数据

对于单个字段，开发者可以使用数据绑定的方法进行信息更新。绑定的数据除了在加载的时候可以更新，也可以在JS主程序中以函数形式进行更新，更新同样可以反应到界面上被绑定的数据中。

3. 条件渲染与列表（循环）渲染

条件渲染适用于有意外情况提示的页面（如无法加载列表或详情时，做出提示等等）。它的渲染带有触发条件，即符合条件时渲染这个页面，否则忽略或渲染另一端代码。两个花括号所包含的判断条件中的变量于主程序JS代码中的data中声明。将同一元素渲染代码进行集合。循环的数据可以通过数组的方式写入data中供WXML访问。渲染完毕后，渲染判断条件的变动可以影响界面变动。

4. 模板与引用

WXML支持使用模板与引用减少代码体积。模板是在WXML代码中对相同的代码进行复用的方式。可以将多个模板写入至同一个文件，并使用import在其他文件中进行引用。如果需要整个页面引用，需要使用include。

5. 样式

通过WXSS样式表，开发者可以定义WXML中的元素样式。WXSS与CSS代码一样，可以直接使用选择器选择元素，在WXML中也可以直接定义元素的id和class以便于在WXSS文件中进行样式定义。

6. 用户操作与事件响应

由于微信使用的不是HTML，所以也不能通过添加超链接的方式来检测用户的点击事件。对于需要监听点击事件的元素，应该在WXML中使用bindtap属性或catchtap属性进行绑定。除了点击一次，微信也提供按住、开始触摸、松手等事件响应。在WXML中绑定好一个事件之后，就能在主程序中使用。其他的API中也有相应的事件，这些事件乐意在微信小程序的官方文档中查阅到。当需要在小程序的页面间进行跳转时，应该使用wx.navigateTo()方式。注意，有关于页面层级跳转，微信将层级跳转限制在5层。开发时注意不要超过相应限制。

数据缓存

每个微信小程序都可以有自己的本地缓存，可以通

过 `wx.setStorage` (`wx.setStorageSync`)、`wx.getStorage` (`wx.getStorageSync`)、`wx.clearStorage` (`wx.clearStorageSync`)

可以对本地缓存进行设置、获取和清理。本地缓存最大为**10MB**。

✓ 数据

数据缓存

- wx.setStorage
- wx.setStorageSync
- wx.getStorage
- wx.getStorageSync
- wx.getStorageInfo
- wx.getStorageInfoSync
- wx.removeStorage
- wx.removeStorageSync
- wx.clearStorage
- wx.clearStorageSync

wx.setStorage(OBJECT) 将数据存储在本地的缓存中指定的 key 中，会覆盖掉原来该 key 对应的内容，这是一个异步接口。

OBJECT参数说明：

参数	类型	必填	说明
key	String	是	本地缓存中的指定的 key
data	Object/String	是	需要存储的内容
success	Function	否	接口调用成功的回调函数
fail	Function	否	接口调用失败的回调函数
complete	Function	否	接口调用结束的回调函数（调用成功、失败都会执行）

示例代码

```
wx.setStorage({
  key: "key"
  data: "value"
})
```

异步调用

Promise

```

// 定义
network.doRequest = function(options){
    return new Promise((resolve, reject) => {

        doSth((data) => {
            if(success){
                resolve(data);
            }else{
                reject(new Error('出错啦! '));
            }
        });
    });
};

// 使用
network.doRequest(options).then((data) => {
    //success
}).catch((err) => {
    //err
});

// 多个异步过程连续调用
let a = (data) => {
    return network.doRequest(options);
};

let b = (data) => {
    return network.doRequest(options);
};

let c = (data) => {
    return network.doRequest(options);
};

// 串行
Promise.resolve()
    .then(a)
    .then(b)
    .then(c)
    .then((data) => {
        // a, b and c all success.
    }).catch((err) => {
        // some method got error.
    }).then(() => {
        // always/finally
    });

// 并行
Promise.all([a(), b(), c()]).then((dataA, dataB, dataC) => {
    // a, b and c all success.
}).catch((err) => {
    // some method got error.
})

```

小程序设计基础

尺寸单位

- rpx (responsive pixel) : 可以根据屏幕宽度进行自适应。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750rpx = 375px = 750物理像素，1rpx = 0.5px = 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	1rpx = 0.42px	1px = 2.34rpx
iPhone6	1rpx = 0.5px	1px = 2rpx
iPhone6 Plus	1rpx = 0.552px	1px = 1.81rpx

- rem (root em) : 规定屏幕宽度为20rem; 1rem = (750/20)rpx 。

建议：开发微信小程序时设计师可以用 iPhone6 作为视觉稿的标准。注意：在较小的屏幕上不可避免的会有一些毛刺，请在开发时尽量避免这种情况。

注意事项

- `data` 的更新一定要在 `setData` 方法中操作。底层做了 `data` 与组件之间的绑定，只有在 `setData` 中操作才会自动更新到组件中；
- 基本组件默认有bindtap属性，可直接使用，`<button>` 只提供了三个基本样式 `default`、`primary`、`warn`，并且有 `border`；

CSS

- 由于小程序当前的程序包体积限制在1M以内，所以尽量控制图片的使用。提示性图标的使用多采用iconfont方案替代；
 - iconfont优势
 1. 灵活性，改变图标的颜色，背景色，大小都非常简单
 2. 兼容性，兼容所有流行的浏览器，不仅h5可以使用iconfont，app也可以使用iconfont，关于这方面可以查看其它线上分享
 3. 扩展性，替换图标很方便，新增图标也非常简单，也不需要考虑图标合并的问题，图片方案需要css sprite
 4. 高效性，iconfont有矢量特性，没有图片缩放消耗高
 - 在使用上字体文件和普通的静态资源一样，既可以外链也可以内链，并且字体文件也可以使用gzip压缩；
 - 在移动端上，可以只使用truetype类型，非常灵活小巧
 - 现在很多项目已经在使用iconfont，先不说国外，比如国内，阿里巴巴各个平台（不仅pc，h5，还有app）已经全面使用iconfont，并且阿里巴巴还搭建了一个线上iconfont站点，这是一个很完善的站点，上面有阿里几个主要业务的图标资源库，也可以让用户自己制作图标，完善用户自身的图标库，让用户之间可以共享形成生态，同时站点的使用说明也非常完整，从图标设计，iconfont制作和iconfont使用（里面包含了各个平台的使用方法）都有很完善的说明
 - iconfont缺点
 1. 制作iconfont需要svg设计稿，对于开发来说没有图片来的方便
 2. iconfont有些特有的问题，详情可参考@font-face and performance，不过许多问题在移动端是不存在的
 - [iconfont简介](#) [阿里巴巴iconfont](#) [富途iconfont](#) [富途iconfont CSS](#)
- [position\(relative/absolute\)](#)

问题整理

- `this.setData({ test: 1 })` 调用结束后，`this.data.test` 才会访问到正确的赋值结果
- Android输入框内输入字符，点击键盘上的完成小程序闪退
- 点击输入框获取焦点时，原有内容会抖动一下
- 模态弹窗的确定取消按钮回调结果iOS和Android返回的类型不同


```
wx.showModal({
  title: '提示',
  content: '这是一个模态弹窗',
  success: function(res) {
    if (res.confirm) {
      console.log('用户点击确定')
    }
  }
})
```

- [微信公开课北京](#)

ST环境搭建¹

- PackageControl
- JavaScriptNext
- Emmet
- Git
- JsFormat
- DocBlockr
- AutoFileName
- ColorPicker
- JSHint

- [微信小程序代码提示插件](#)

- [ST插件推荐](#)

参考²

- [微信小程序开发文档](#)
- [微信小程序设计指南](#)
- [微信小程序开发工具下载](#)
- [官方体验Demo](#)
- [微信小程序『官方示例代码』浅析：上](#)
- [微信小程序『官方示例代码』浅析：下](#)
- [微信基础样式库参考](#)
- [We-UI_Sketch组件库](#)
- [We-UI_PS组件库](#)
- [ECMAScript 6入门](#)
- [MDN JS Docs](#)
- [JS DevDocs](#)
- [CSS Docs](#)
- [Javascript继承机制的设计思想](#)
- [Javascript 面向对象编程（一）：封装](#)
- [Javascript 面向对象编程（二）：构造函数的继承](#)
- [Javascript 面向对象编程（三）：非构造函数的继承](#)

1. Sublime Text 环境搭建 ↩

2. 参考 ↩