

微信自己封装了一套框架，开发了一个小程序开发者工具，专门用于小程序开发。

框架

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。

组成

- 响应的数据绑定
- 页面管理
- 基础组件
- 丰富的API

框架模式

- 视图层
- 逻辑层
- 模型层

文件结构

小程序包含一个描述整体程序的app和多个描述各自页面的page

小程序主体由三个文件组成，必须在项目根目录

文件	必填	作用
app.js	是	小程序逻辑
app.json	是	小程序配置
app.wxss	否	公共样式表

小程序的页面由四个文件组成

文件	必填	作用
ic	是	页面逻辑

属性	是否必填	描述
wxml	是	页面结构
wxss	否	页面样式表
json	否	页面配置

配置

全局配置 - 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

页面配置 - index.json

页面配置只能配置window下的配置，且不必使用window建

属性	类型	默认值	描述
navigationBarBackgroundColor	HexColor	#000000	导航栏背景颜色，如"#000000"
navigationBarTextStyle	String	white	导航栏标题颜色，仅支持 black、white
navigationBarTitleText	String		导航栏标题文字内容
backgroundColor	HexColor	#ffffff	窗口的背景色
backgroundTextStyle	String	dark	下拉背景字体、loading 图的背景色，仅支持 dark、light
enablePullDownRefresh	Boolean	false	是否开启下拉刷新
onReachBottomDistance	Number	50	页面上拉触底事件触发时距当前底部的高度

逻辑层

`App()` 函数用来注册一个小程序。接受一个 `object` 参数，其指定小程序的生命周期函数等。

`object` 参数：

- `onLaunch`
- `onShow`
- `onHide`
- `onError`
- 其他

`App()` 必须在 `app.js` 中注册，且不能注册多个。

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。

不要在 `onLaunch` 的时候调用 `getCurrentPages()`，此时 `page` 还没有生成。

通过 `getApp()` 获取实例之后，不要私自调用生命周期函数。

`Page()` 函数用来注册一个页面。接受一个 `object` 参数，其指定页面的初始数据、生命周期函数、事件处理函数等。

`object` 参数说明

- `onReady`
- `onShow`
- `onHide`
- `onUnload`
- `onPullDownRefresh`
- `onReachBottom`
- `onShareAppMessage`
- `onPageScroll`
- `setData`

路由

文件作用域

在 JavaScript 文件中声明的变量和函数只在该文件中有效；不同的文件中可以声明相同名字的变量和函数，不会互相影响。

通过全局函数 `getApp()` 可以获取全局的应用实例，如果需要全局的数据可以在 `App()` 中设置

模块化

我们可以将一些公共的代码抽离成为一个单独的 js 文件，作为一个模块。模块只有通过 `module.exports` 或者 `exports` 才能对外暴露接口。

需要注意的是：

`exports` 是 `module.exports` 的一个引用，因此在模块里边随意更改 `exports` 的指向会造成未知的错误。

所以我们更推荐开发者采用 `module.exports` 来暴露模块接口，除非你已经清晰知道这两者的关系。

小程序目前不支持直接引入 `node_modules`，开发者需要使用到 `node_modules` 时候建议拷贝出相关的代码到小程序的目录中。

tip: `exprots` 暂不支持绝对路径

视图层

WXML：框架设计的一套标签语言，结合基础组件、事件系统，可以构建出页面的结构。

数据绑定： `{{message}}`

列表渲染： `<view wx:for="{{array}}">{{item}}</view>`

条件渲染：

`<view wx:if="{{view == 'WEBVIEW'}}"> WEBVIEW </view>`

`<view wx:elif="{{view == 'APP'}}"> APP </view>`

`<view wx:else="{{view == 'MINA'}}"> MINA </view>`

模板：

`<template name="staffName">`

`<view>`

`FirstName: {{firstName}}, LastName: {{lastName}}`

`</view>`

`</template>`

`<template is="staffName" data="{{...staffA}}"></template>`

`<template is="staffName" data="{{...staffB}}"></template>`

`<template is="staffName" data="{{...staffC}}"></template>`

事件：

```
<view bindtap="add"> {{count}} </view>
```

引用

WXS：小程序的一套脚本语言，结合 WXML，可以构建出页面的结构。

模块

变量

注释

运算符

语句

数据类型

基础类库

WXSS：一套样式语言，用于描述 WXML 的组件样式。

尺寸单位：rpx

样式导入

内联表达式

选择器

全局样式与局部样式

组件

基础库：小程序的能力需要微信客户端来支撑，每一个基础库都只能在对应的客户端版本上运行，高版本的基础库无法兼容低版本的微信客户端。

兼容

运行机制

性能

组件

视图容器

基础内容

表单组件

导航
媒体组件
地图
画布
开放数据

API

网络
媒体
文件
数据缓存
位置
设备
界面
WXML节点信息
第三方平台
开发接口
数据
扩展接口
调试接口

工具

小小的实例

1. 配置app.json
2. component
3. js data
4. wxml
5. wxcss
6. 页面之间的链接
7. 审核上线

