

网媒组

小程序开发介绍



模拟器 编辑器 调试器 云开发

小程序模式

普通编译

编译

预览

真机调试

切后台

清缓存

上传

版本管理

社区

素材管理

详情

iPhone 6

100%

WiFi



app.json

```
1  {
2    "pages": [
3      "pages/index/index",
4      "pages/logs/logs",
5      "pages/my/my",
6      "pages/recruit/form",
7      "pages/recruit/result",
8      "pages/resume/resume",
9      "pages/room/room",
10     "pages/room/register",
11     "pages/room/booking"
12   ],
13   "window": {
14     "backgroundTextStyle": "light",
15     "navigationBarBackgroundColor": "#fff",
16     "navigationBarTitleText": "西电华小为",
17     "navigationBarTextStyle": "black"
18   },
19   "tabBar": {
20     "color": "black",
21     "selectedColor": "black",
22     "backgroundColor": "#80D5D4",
23     "list": [
24       {
25         "pagePath": "pages/index/index",
26         "iconPath": "public/img/tabbar/index.png",
27         "selectedIconPath": "public/img/tabbar/index_selc.png",
28         "text": "首页"
29       },
30       {
31         "pagePath": "pages/my/my",
32         "iconPath": "public/img/tabbar/my.png",
33         "selectedIconPath": "public/img/tabbar/my_selc.png",
34         "text": "我的"
35       }
36     ]
37   },
38   "plugins": {
39   }
40 }
41 }
```

/app.json 958 B

JSON

Console Sources Network Security AppData Audits Sensor Storage Trace Wxml

top

Filter

Default levels

3 items hidden by filters

代码结构

● App

JS WXSS JSON

● Page

JS WXSS JSON WXML

● project.config.json

```
▶ 文件夹 pages
▶ 文件夹 public
▶ 文件夹 utils
JS app.js
{ } app.json
WXSS app.wxss
{ } project.config.json
```

App - 全局配置

- **app.js**


小程序启动

- **app.wxss**

全局样式

- **app.json**

全局配置属性



- ▶ 文件夹 pages
- ▶ 文件夹 public
- ▶ 文件夹 utils
- JS app.js
- { } app.json
- WXSS app.wxss
- {@} project.config.json

app.json

对微信小程序进行全局配置，决定页面文件的路径、窗口表现、设置网络超时时间、设置多 tab 等。

● pages

注册小程序的所有页面

● window

默认的窗口表现，如navigationBarBackgroundColor、navigationBarTitleText等

● tabBar

配置tab栏的属性，如color、selectedColor、list、position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
}
```

app.wxss

小程序的全局样式

- 会继承到每个page上
- 常常利用import引入UI库的WXSS



The screenshot shows a code editor window titled 'app.wxss' with a close button. The code inside is as follows:

```
1  /**app.wxss**/  
2  @import "../public/css/weui.wxss";  
3
```

app.js

对小程序的启动脚本

● 注册app实例

可在page中利用getApp()获取此实例

● 实现生命周期函数

特指小程序（App）的生命周期，与页面（page）的生命周期应区分开。
如onLaunch、onShow、onHide、onError等

●.globalData

存储公共数据，一般仅存储一些常量，如SecretKey、URL Prefix等。

```
1 //app.js
2 App({
3   onLaunch: function () {
4     // 展示本地存储能力
5     var logs = wx.getStorageSync('logs') || []
6     logs.unshift(Date.now())
7     wx.setStorageSync('logs', logs)
8
9     // 登录
10    wx.login({
11      success: res => {
12        // 发送 res.code 到后台换取 openId, sessionKey, unionId
13      }
14    })
15    // 获取用户信息
16    wx.getSetting({
17      success: res => {
18        if (res.authSetting['scope.userInfo']) {
19          // 已经授权，可以直接调用 getUserInfo 获取头像昵称，不会弹框
20          wx.getUserInfo({
21            success: res => {
22              // 可以将 res 发送给后台解码出 unionId
23              this.globalData.userInfo = res.userInfo
24
25              // 由于 getUserInfo 是网络请求，可能会在 Page.onLoad 之后才返回
26              // 所以此处加入 callback 以防止这种情况
27              if (this.userInfoReadyCallback) {
28                this.userInfoReadyCallback(res)
29              }
30            }
31          })
32        }
33      }
34    })
35  },
36  globalData: {
37    userInfo: null
38  }
39 })
```

Page – 页面配置

- **page.js**

对应JS，逻辑层实现，涉及生命周期、数据处理、页面跳转、组件调用等

- **page.wxml**

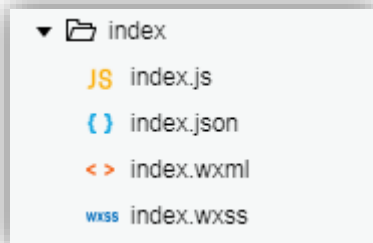
对应HTML，标签语言，构建视图层的结构，利用逻辑层数据渲染页面

- **page.wxss**

对应CSS，页面样式

- **page.json**

页面配置属性，覆盖app.json的相同配置项，只在本页面有效。



page.js

逻辑层实现，涉及生命周期、数据处理、页面跳转、组件调用等

● 数据

利用data存储当前page的数据，供逻辑层处理、供视图层渲染。

逻辑层：`this.setData({ text: 'abc' })` 视图层：`{{text}}`

● 生命周期

框架提供的生命周期钩子，供开发者在page生命周期的不同阶段进行相应操作。

● wx.xxx API

小程序开发框架提供丰富的微信原生 API，可以方便的调起微信提供的能力，如微信登录、页面跳转、本地存储、HTTP请求等。

```
1 // pages/test/test.js
2 Page({
3
4   /**
5    * 页面的初始数据
6    */
7   data: {
8     text: 'hello world'
9   },
10
11   /**
12    * 生命周期函数--监听页面加载
13    */
14   onLoad: function (options) {
15     console.log('[onLoad] The test page is loaded.');
```

page.js

逻辑层实现，涉及生命周期、数据处理、页面跳转、组件调用等

● 数据

利用data存储当前page的数据，供逻辑层处理、供视图层渲染。

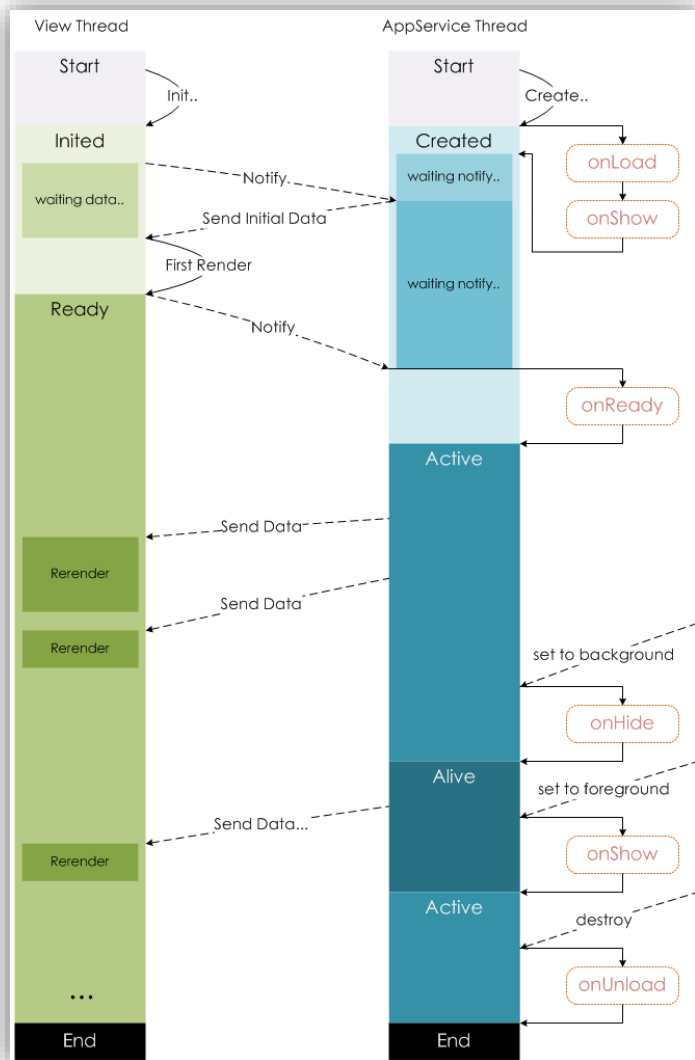
逻辑层：`this.setData({ text: 'abc' })` 视图层：`{{text}}`

● 生命周期

框架提供的生命周期钩子，供开发者在page生命周期的不同阶段进行相应操作。

● wx.xxx API

小程序开发框架提供丰富的微信原生 API，可以方便的调起微信提供的能力，如微信登录、页面跳转、本地存储、HTTP请求等。



page.wxml

对应HTML，标签语言，构建视图层的结构，利用逻辑层数据渲染页面

● 数据绑定

数据绑定使用 Mustache 语法（双大括号）将变量包起来，数据均来自于page中的data中的键值对。

● 列表渲染 & 条件渲染 & 事件绑定

利用data中的数组for遍历，重复渲染；

使用 wx:if="{{condition}}" 来判断是否需要渲染；

最基础的绑定：bindtap。

● 基础组件与自定义组件

所有的组件都以<>标签的方式使用，配合JS使用；

基础组件就是内置组件，div->view，span->text，button，input等；

自定义组件是需要掌握的重要的开发能力。

```
1  <!--index.wxml-->
2  <view class="container">
3    <view class="userinfo">
4      <button wx:if="{{!hasUserInfo && canIUse}}" open-type="getUserInfo" bindgetuserinfo="getUserInfo"> 获取头像昵称 </button>
5      <block wx:else>
6        <image bindtap="bindViewTap" class="userinfo-avatar" src="{{userInfo.avatarUrl}}" mode="cover"></image>
7        <text class="userinfo-nickname">{{userInfo.nickName}}</text>
8      </block>
9    </view>
10   <view class="usermotto">
11     <text class="user-motto">{{motto}}</text>
12   </view>
13 </view>
```

page.wxss

页面样式

page.json

页面配置属性

- 覆盖app.json的相同配置项
- 只在本页面有效

project.config.js

- 这是开发工具的配置文件
- 暴露APPID , 加入.gitingore !

```
1 {
2   "description": "项目配置文件",
3   "packOptions": {
4     "ignore": []
5   },
6   "setting": {
7     "urlCheck": true,
8     "es6": true,
9     "postcss": true,
10    "minified": true,
11    "newFeature": true
12  },
13  "compileType": "miniprogram",
14  "libVersion": "2.0.4",
15  "appid": "wx2[REDACTED]945fbae3",
16  "projectname": "hello-weapp",
17  "debugOptions": {
18    "hidedInDevtools": []
19  },
20  "isGameTourist": false,
21  "condition": {
22    "search": {
23      "current": -1,
24      "list": []
25    },
26    "conversation": {
27      "current": -1,
28      "list": []
29    },
30    "game": {
31      "currentL": -1,
32      "list": []
33    },
34    "miniprogram": {
35      "current": -1,
36      "list": []
37    }
38  }
39 }
```

代码结构

● App

JS WXSS JSON

● Page

JS WXSS JSON WXML

● project.config.json

- ▼ public
 - ▶ css
 - ▶ img

公共资源

公共方法

- ▼ utils
 - ▶ leancloud
 - JS book.js
 - JS recruit2018.js
 - JS to.js
 - JS util.js

- ▶ pages
- ▶ public
- ▶ utils
- JS app.js
- { } app.json
- WXSS app.wxss
- {@} project.config.json

API使用演示

□ 生命周期

□ 页面跳转

□ showToast

□ Tabbar

□ 网络请求

□ 下拉刷新

□ 列表循环