

## 【摘要】微信小程序初步探究简单整理

下面是我初步探究所做的简单整理。

### 开发工具下载

## 目录结构

我们在微信提供的开发工具中新建一个项目，开发工具会给我们建立一个最基本的目录结构和示例 Demo 代码，结构如下：

```
▼ pages
  ▼ index
    index.js
    index.wxml
    index.wxss
  ▼ logs
    logs.js
    logs.json
    logs.wxml
    logs.wxss
  ▼ utils
    util.js
  app.js
  app.json
  app.wxss
```

结构很简单，程序主体部分由 `app.js`, `app.json`, `app.wxss` 三个文件组成，而且必须放在项目的根目录。页面由四个文件组成，分别是：

文件类型	作用	必填
js	页面逻辑	yes
wxml	页面结构	yes
wxss	页面样式	no
json	页面配置	no

## 配置

- 使用 `app.json` 文件来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。
- 每一个小程序页面也可以使用 `.json` 文件来对本页面的窗口表现进行配置。页面的配置比 `app.json` 全局配置简单得多，只是设置 `app.json` 中的 `window` 配置项的内容，页面中配置项会覆盖 `app.json` 的 `window` 中相同的配置项。

## 逻辑层

### 1.注册程序

`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。

示例：

```
App({
  onLaunch: function() {
    // Do something initial when launch.
  },
  onShow: function() {
    // Do something when show.
  },
})
```

```
onHide: function() {  
    // Do something when hide.  
},  
globalData: 'I am global data'  
})
```

## 2.注册页面

**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。

```
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.  
  },  
  onPullDownRefresh: function() {  
    // Do something when pull down  
  },  
})
```

```
// Event handler.  
viewTap: function() {  
  this.setData({  
    text: 'Set some data for updating view.'  
  })  
}  
});
```

### 3.模块化

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。

### 4.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，如获取用户信息，本地存储，支付功能等。

### 5.ES6 转 ES5

微信小程序运行在三端：`iOS`、`Android` 和用于调试的开发者工具

- 在 iOS 上，小程序的 javascript 代码是运行在 JavaScriptCore 中
- 在 Android 上，小程序的 javascript 代码是运行在 X5 内核中
- 在 开发工具上，小程序的 javascript 代码是运行在 nwjs (chrome) 中

虽然三个运行环境在大部分情况下是相似的，但是还有一些细微的区别，为了帮助开发者解决这种区别带来的困扰，开发工具会自动帮助开发者将 ES6 的代码转为 ES5 的代码。

对于使用其他构建工具的开发者，可以在项目也开中关掉这个功能，使用自己的构建和转码工具。

## 视图层

和我之前对微信小程序的直觉不同，微信小程序不支持也不兼容 HTML，而是微信全新定义的规范，它的视图文件的后缀名为 .wxml，是基于 xml 进行的扩展，其样式表文件也并非 CSS，而是 .wxss，兼容受限的部分 CSS 写法。

视图渲染时，采用了类似单向数据绑定的方式进行数据绑定，WXML 中的动态数据均来自对应 Page 的 data。使用 Mustache 语法（双大括号）将变量包起来：

```
<view> {{ message }} </view>
```

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。支持条件渲染、列表渲染、模板、事件

WXSS(WeiXin Style Sheets)是一套样式语言，用于描述 WXML 的组件样式。

WXSS 用来决定 WXML 的组件应该怎么显示。

WXSS 具有 CSS 大部分特性。同时为了更适合开发微信小程序，我们对 CSS 进行了扩充以及修改。

与 CSS 相比我们扩展的特性有：

- 尺寸单位
- 样式导入

框架为开发者提供了一系列基础组件，开发者可以通过组合这些基础组件进行快速开发。[组件文档](#)

微信小程序官方文档[传送门](#)