

微信小程序学习报告

1. 申请账号

点击 <https://mp.weixin.qq.com/wxopen/waregister?action=step1> 根据指引填写信息和提交相应的资料

小程序注册

① 账号信息

② 邮箱激活

③ 信息登记

每个邮箱只能申请一个小程序

已有微信小程序？立即登录

邮箱

作为登录账号，请填写未被微信公众平台注册、未被微信开放平台注册、未被个人微信号绑定的邮箱

密码

字母、数字或者英文符号，最短8位，区分大小写

确认密码

请再次输入密码

验证码

换一张

☐ 我已阅读并同意《微信公众平台服务协议》及《微信小程序平台服务条款》

注册

<https://blog.csdn.net/zheWILL>

2. 安装开发模板

下载开发工具

微信公众平台 | 小程序

介绍 | 设计 | 小程序开发 | 运营 | 数据

简易教程 | 框架 | 组件 | API | 工具 | 腾讯云支持

概览

界面

启动页

菜单栏

工具栏

模拟器

设置

外观设置

编辑设置

代理设置

通知设置

项目页卡

项目设置

域名信息

最新版本下载地址 (1.02.1804090)

windows 64、windows 32、mac

2018.04.09 更新日志

1. 修复 1.02.180408Q 引入的页面白屏的问题 [详情](#)

2018.04.08 更新日志

1. A 新增 编辑器文件标签打开策略设置

2. A 新增 iPhone X 刘海效果

3. A 新增 App.onPageNotFound 的调试支持 [详情](#)

4. A 新增 远程调试带插件的小程序支持

5. A 新增 project.config.json 的 ignore 字段 [详情](#)

6. A 新增 项目设置中展示插件信息、分包大小

7. A 新增 自定义预处理命令 [详情](#)

8. A 新增 支持分包中使用插件

<https://blog.csdn.net/zheWILL>

项目 文件 编辑 工具 界面 设置 微信开发者工具

模拟器 编辑器 调试器

普通编译 编译 预览 远程调试 切后台 清缓存

iPhone 5 100% WiFi

15:44 96% WeChat

pages

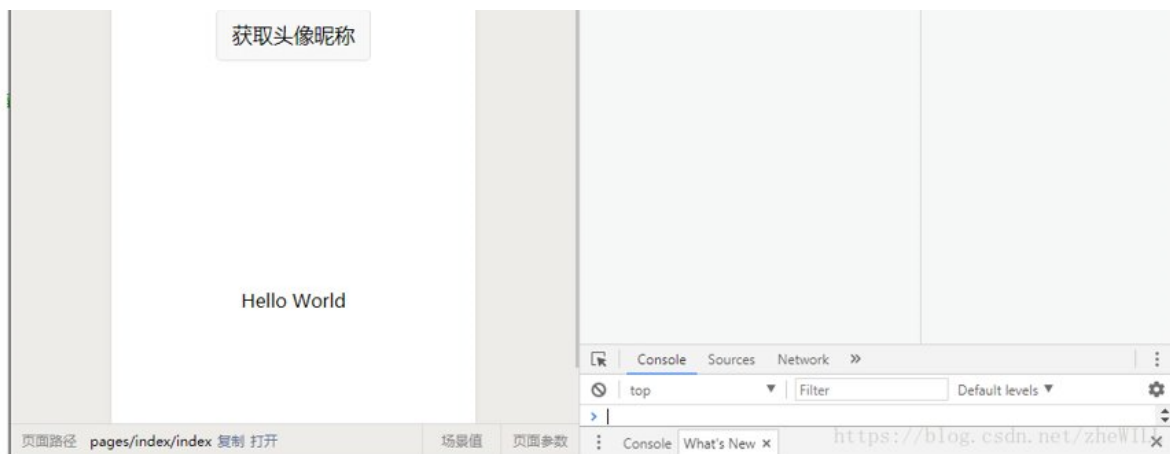
utils

JS app.js

app.json

app.wxss

project.config.json



小程序配置 app.json

app.json 是对当前小程序的全局配置，包括了小程序的所有页面路径、界面表现、网络超时时间、底部 tab 等。

```
{
  "pages": [
    "pages/index/index",
    "pages/logs/logs"
  ],
  "window": {
    "backgroundTextStyle": "light",
    "navigationBarBackgroundColor": "#fff",
    "navigationBarTitleText": "WeChat",
    "navigationBarTextStyle": "black"
  }
}
```

<https://blog.csdn.net/zheWILL>

3.WXML 模板

从事过网页编程的人知道，网页编程采用的是 HTML + CSS + JS 这样的组合，其中 HTML 是用来描述当前这个页面的结构，CSS 用来描述页面的样子，JS 通常是用来处理这个页面和用户的交互。

```
[html]
1. <view class="container">
2.   <view class="userinfo">
3.     <button wx:if="{{!hasUserInfo && canIUse}}"> 获取头像昵称 </button>
4.     <block wx:else>
5.       <image src="{{userInfo.avatarUrl}}" background-size="cover"></image>
6.       <text class="userinfo-nickname">{{userInfo.nickName}}</text>
7.     </block>
8.   </view>
9.   <view class="usermotto">
10.    <text class="user-motto">{{motto}}</text>
11.  </view>
12. </view>
```

4.WXSS 样式

WXSS 具有 CSS 大部分的特性，小程序在 WXSS 也做了一些扩充和修改。

1. 新增了尺寸单位。在写 CSS 样式时，开发者需要考虑到手机设备的屏幕会有不同的宽度和设备像素比，采用一些技巧来换算一些像素单位。WXSS 在底层支持新的尺寸单位 rpx，开发者可以免去换算的烦恼，只要交给小程序底层来换算即可，由于换算采用的浮点数运算，所以运算结果会和预期结果有一点点偏差。
2. 提供了全局的样式和局部样式。和前边 app.json, page.json 的概念相同，你可以写一个 app.wxss 作为全局样式，会作用于当前小程序的所有页面，局部页面样式 page.wxss 仅对当前页面生效。
3. 此外 WXSS 仅支持部分 CSS 选择器

5.JS 交互逻辑

一个服务仅仅只有界面展示是不够的，还需要和用户做交互：响应用户的点击、获取用户的位置等等。在小程序里边，我们就通过编写 JS 脚本文件来处理用户的操作。

```
<view>{{ msg }}</view>
<button bindtap="clickMe">点击我</button>
```

点击 button 按钮的时候，我们希望把界面上 msg 显示成 "Hello World"，于是我们在 button 上声明一个属性: bindtap，在 JS 文件里边声明了 clickMe 方法来响应这次点击操作：

```
Page({
  clickMe: function() {
    this.setData({ msg: "Hello World" })
  }
})
```

6.小程序的启动

微信客户端在打开小程序之前，会把整个小程序的代码包下载到本地。

紧接着通过 app.json 的 pages 字段就可以知道你当前小程序的所有页面路径：

```
{
  "pages": [
    "pages/index/index",
    "pages/logs/logs"
  ]
}
```

这个配置说明在 QuickStart 项目定义了两个页面，分别位于 pages/index/index 和 pages/logs/logs 目录。而写在 pages 字段的第一个页面就是这个小程序的首页(打开小程序看到的第一个页面)。

于是微信客户端就把首页的代码装载进来，通过小程序底层的一些机制，就可以渲染出这个首页。

小程序启动之后，在 app.js 定义的 App 实例的 onLaunch 回调会被执行：

```
App({
  onLaunch: function () {
    // 小程序启动之后 触发
  }
})
```

接下来我们简单看看小程序的一个页面是怎么写的。

7.程序与页面

你可以观察到 pages/logs/logs 下其实是包括了4种文件的，微信客户端会先根据 logs.json 配置生成一个界面，顶部的颜色和文字你都可以在这个 json 文件里边定义好。紧接着客户端就会装载这个页面的 WXML 结构和 WXSS 样式。最后客户端会装载 logs.js，你可以看到 logs.js 的大体内容就是：

```
Page({
  data: { // 参与页面渲染的数据
    logs: []
  },
  onLoad: function () {
    // 页面渲染后 执行
  }
})
```

Page 是一个页面构造器，这个构造器就生成了一个页面。在生成页面的时候，小程序框架会把 data 数据和 index.wxml 一起渲染出最终的结构，于是就得到了你看到的小程序的样子。

在渲染完界面之后，页面实例就会收到一个 onLoad 的回调，你可以在这个回调处理你的逻辑。。

8. 组件

小程序提供了丰富的基础组件给开发者，开发者可以像搭积木一样，组合各种组件拼合成自己的小程序。

就像 HTML 的 div, p 等标签一样，在小程序里边，你只需要在 WXML 写上对应的组件标签名字就可以把该组件显示在界面上，例如，你需要在界面上显示地图，你只需要这样写即可：

```
<map></map>
```

使用组件的时候，还可以通过属性传递值给组件，让组件可以以不同的状态去展现，例如，我们希望地图一开始的中心的经纬度是广州，那么你需要声明地图的 longitude(中心经度) 和 latitude(中心纬度) 两个属性：

```
<map longitude="广州经度" latitude="广州纬度"></map>
```

组件的内部行为也会通过事件的形式让开发者可以感知，例如用户点击了地图上的某个标记，你可以在 js 编写 markertap 函数来处理：

```
<map bindmarkertap="markertap" longitude="广州经度" latitude="广州纬度"></map>
```

当然你也可以通过 style 或者 class 来控制组件的外层样式，以便适应你的界面宽度高度等等。

API

为了让开发者可以很方便的调起微信提供的能力，例如获取用户信息、微信支付等等，小程序提供了很多 API 给开发者去使用。

要获取用户的地理位置时，只需要：

```
wx.getLocation({
  type: 'wgs84',
  success: (res) => {
    var latitude = res.latitude // 经度
    var longitude = res.longitude // 纬度
  }
})
```

调用微信扫一扫能力，只需要：

```
wx.scanCode({
  success: (res) => {
    console.log(res)
  }
})
```

需要注意的是：多数 API 的回调都是异步，你需要处理好代码逻辑的异步问题。