

小程序入门

目录

COMPANY

01 什么是小程序

02 申请自己的小程序

03 开发者工具和开发文档

第一部分

什么是小程序

微信官方文档

小程序是一种全新的连接用户与服务的方式，它可以在微信内被便捷地获取和传播，同时具有出色的使用体验。

小程序提供了一个简单、高效的应用开发框架和丰富的组件及API，帮助开发者在微信中开发具有原生 APP 体验的服务。

① 速度快

无需下载安装，加载速度快于HTML5，微信登录，随时可用

② 无需适配

一次开发，多端兼容，免除了对各种手机机型的适配

③ 社交分享

支持直接或App分享给微信好友和群聊

④ 出色体验

可达到近乎原生App的操作体验和流畅度，在离线状态亦可使用

⑤ 用完即走，随手可得

通过扫码、长按、微信搜索、公众号、好友推荐等方式快速获取服务，用完即走

⑥ 低门槛

已有公众号的组织可快速注册、可快速生成门店小程序

小程序的特点：

- 快速的加载
- 更强大的能力
- 原生的体验
- 易用且安全的微信数据开放
- 高效和简单的开发



小程序文件组成

1、.json 后缀的 JSON 配置文件

JSON 是一种数据格式，并不是编程语言，在小程序中，JSON扮演的静态配置的角色。

JSON 语法:JSON文件都是被包裹在一个大括号中 {}，通过key-value的方式来表达数据，且文件中无法使用注释。

JSON的Key必须包裹在一个双引号中。

举例：app.json

```
app.json
1  {
2    "pages":[
3      "pages/index/index",
4      "pages/logs/logs"
5    ],
6    "window":{
7      "backgroundTextStyle":"light",
8      "navigationBarBackgroundColor": "#fff",
9      "navigationBarTitleText": "WeChat",
10     "navigationBarTextStyle":"black"
11   },
12   "style": "v2",
13   "sitemapLocation": "sitemap.json"
14 }
15
```

2、.wxml 后缀的 WXML 模板文件

WXML 充当的就是类似 HTML 的角色，和 HTML 非常相似，WXML 由标签、属性等等构成。有些标签名字有点不一样，多了一些富有特色的（微信系列）wx:if 这样的属性以及 {{ }} 这样的表达式。WXML 要求标签必须是严格闭合的，没有闭合将会导致编译错误。

举例：index.wxml

```
index.wxml
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>
14
```


3、.wxss 后缀的 WXSS 样式文件

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。新增了自动换算尺寸单位。提供了全局的样式和局部样式。但WXSS 仅支持部分 CSS 选择器。

举例：app.wxss

```
app.wxss  X
1  /**app.wxss**/
2  .container {
3      height: 100%;
4      display: flex;
5      flex-direction: column;
6      align-items: center;
7      justify-content: space-between;
8      padding: 200rpx 0;
9      box-sizing: border-box;
10 }
11
```

4、.js 后缀的 JS 脚本逻辑文件

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。可以在 JS 中调用小程序提供的丰富的 API，利用这些 API 可以很方便的调起微信提供的能力，例如获取用户信息、本地存储、微信支付等。

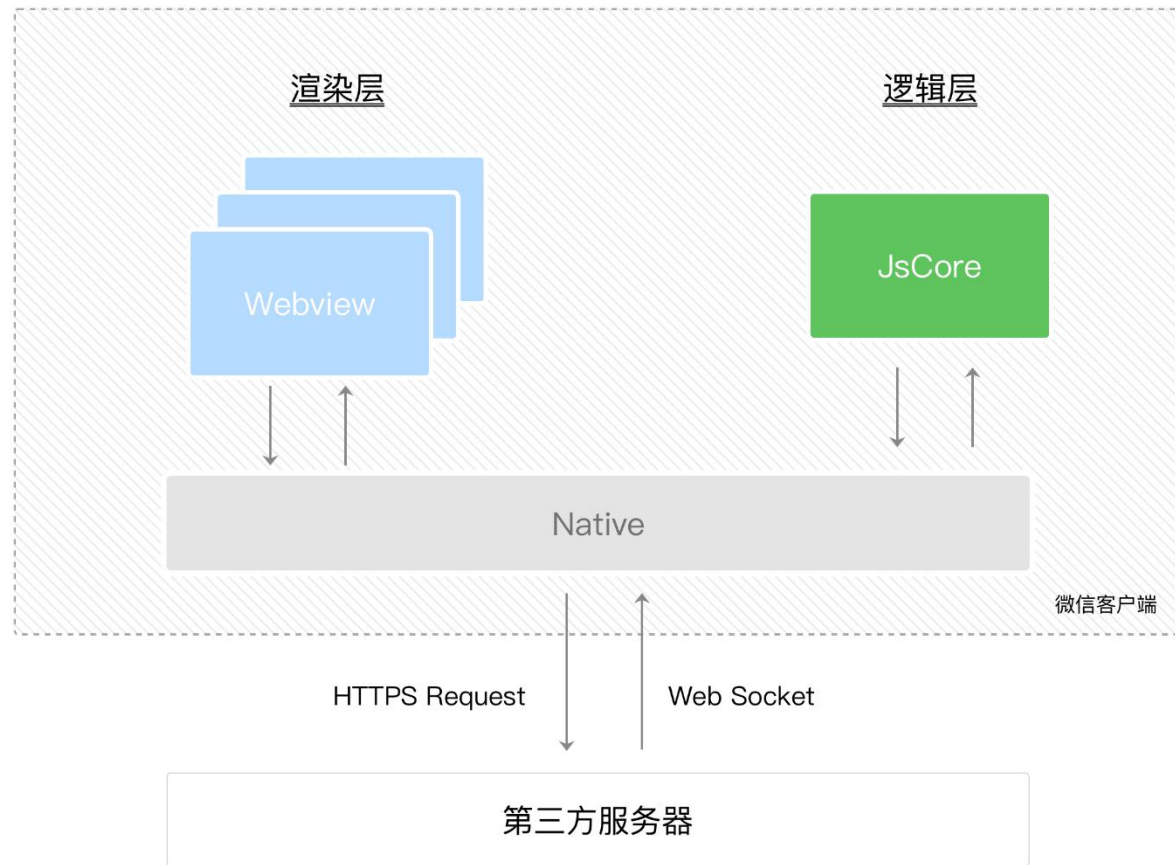
举例：app.js

```
app.js  X
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 })
```

小程序通信模型

小程序的运行环境分成渲染层和逻辑层，其中 WXML 模板和 WXSS 样式工作在渲染层，JS 脚本工作在逻辑层。

小程序的渲染层和逻辑层分别由2个线程管理：渲染层的界面使用了WebView 进行渲染；逻辑层采用 JsCore线程运行JS脚本。一个小程序存在多个界面，所以渲染层存在多个WebView线程，这两个线程的通信会经由微信客户端（下文中也会采用Native来代指微信客户端）做中转，逻辑层发送网络请求也经由Native转发，小程序的通信模型下图所示。

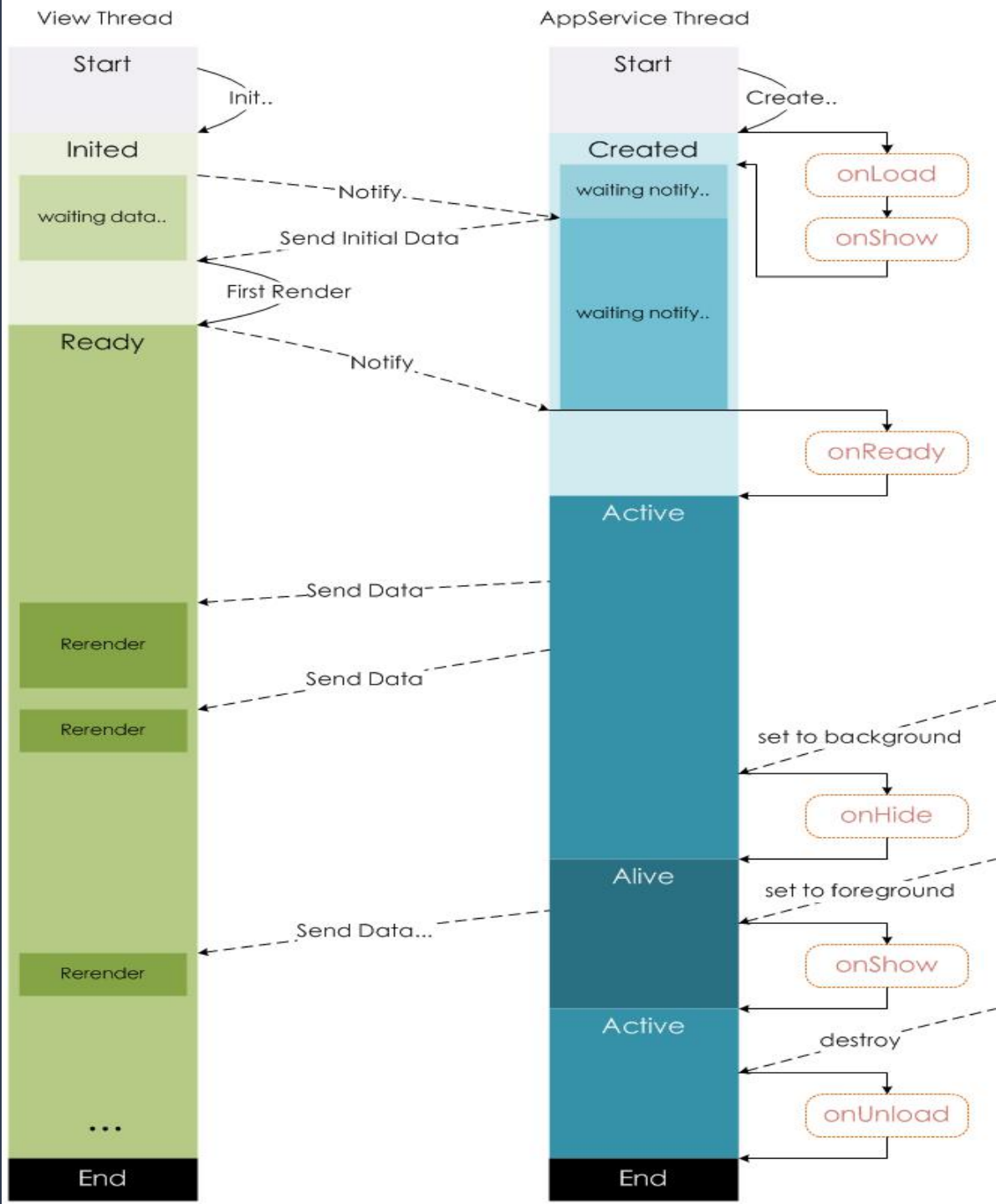


小程序生命周期

onLoad——onShow——onHide——onUnload

加载页面——显示页面——隐藏页面——关闭页面

路由方式	触发时机	路由前页面	路由后页面
初始化	小程序打开的第一个页面		onLoad, onShow
打开新页面	调用 API <code>wx.navigateTo</code> 使用组件 <code><navigator open-type="navigateTo"/></code>	onHide	onLoad, onShow
页面重定向	调用 API <code>wx.redirectTo</code> 使用组件 <code><navigator open-type="redirectTo"/></code>	onUnload	onLoad, onShow
页面返回	调用 API <code>wx.navigateBack</code> 使用组件 <code><navigator open-type="navigateBack"></code> 用户按左上角返回按钮	onUnload	onShow
Tab 切换	调用 API <code>wx.switchTab</code> 使用组件 <code><navigator open-type="switchTab"/></code> 用户切换 Tab		各种情况请参考下表
重启动	调用 API <code>wx.reLaunch</code> 使用组件 <code><navigator open-type="reLaunch"/></code>	onUnload	onLoad, onShow



第二部分

申请小程序

百度搜索微信小程序，找到官网进入



或者输入如下网址进入：
mp.weixin.qq.com/cgi-bin/wx

官网界面如右图示。



官网下拉可以找到开发小程序流程指引，第一步是注册。

接入流程

- 1

注册

在微信公众平台注册小程序，完成注册后可以同步进行信息完善和开发。
- 2

小程序信息完善

填写小程序基本信息，包括名称、头像、介绍及服务范围等。
- 3

开发小程序

完成小程序开发者绑定、开发信息配置后，开发者可下载开发者工具、参考开发文档进行小程序的开发和调试。
- 4

提交审核和发布

完成小程序开发后，提交代码至微信团队审核，审核通过后即可发布（公测期间不能发布）。
- 前往注册

跳转页面后依次完成注册过程即可。完成注册登录后在设置的基本设置项下拉可以找到你的AppID（小程序ID），注意不是Apple ID。

小程序注册

1 帐号信息

2 邮箱激活

3 信息登记

每个邮箱仅能申请一个小程序

已有微信小程序？立即登录

邮箱

594057186@qq.com

作为登录帐号，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

ABDd

换一张

☐ 你已阅读并同意《微信公众平台服务协议》及《微信小程序平台服务条款》

注册

登录后在首页你可以看到开发工具这一栏，点击“普通小程序开发者工具”的蓝色字体链接可以进入。

🏠 首页

📁 管理

版本管理

成员管理

用户反馈

📊 统计

⚙️ 功能

微信搜一搜

客服

订阅消息

页面内容接入

🔧 开发

🌱 成长

小程序评测

违规记录

📢 推广

流量主

广告主

小程序发布流程

1

小程序信息

补充小程序的基本信息，如名称、图标、描述等

已完成

查看详情

小程序开发与管理

添加开发者

开发工具

下载开发工具进行代码的开发和上传：[普通小程序开发工具](#)、[小游戏开发工具](#)

添加开发者

添加开发者，进行代码上传

配置服务器

在开发设置页面查看AppID和AppSecret，配置服务器域名

帮助文档

可以阅读入门介绍（[普通小程序 | 小游戏](#)）、开发文档（[普通小程序 | 小游戏](#)）、设计规范和运营规范

服务平台

提供大量优质的小程序第三方服务，为不同行业的商家提供合适的小程序解决方案。前往 [服务平台](#)

2

版本发布

先提交代码，然后提交审核，审核通过后可发布

前往发布

小程序助手

如果你没有完成注册登录也可以在官网首页下拉找到开发支持一栏，点击“开发者工具”进入。

开发支持

提供一系列工具帮助开发者快速接入并完成小程序开发。



开发文档



开发者工具



设计指南



小程序体验DEMO

概览

界面

设置

代码编辑

小程序调试

开发模式

开发辅助

小程序自动化

工具插件

API 实现差异

下载

稳定版更新日志

预发布版更新日志

开发版更新日志

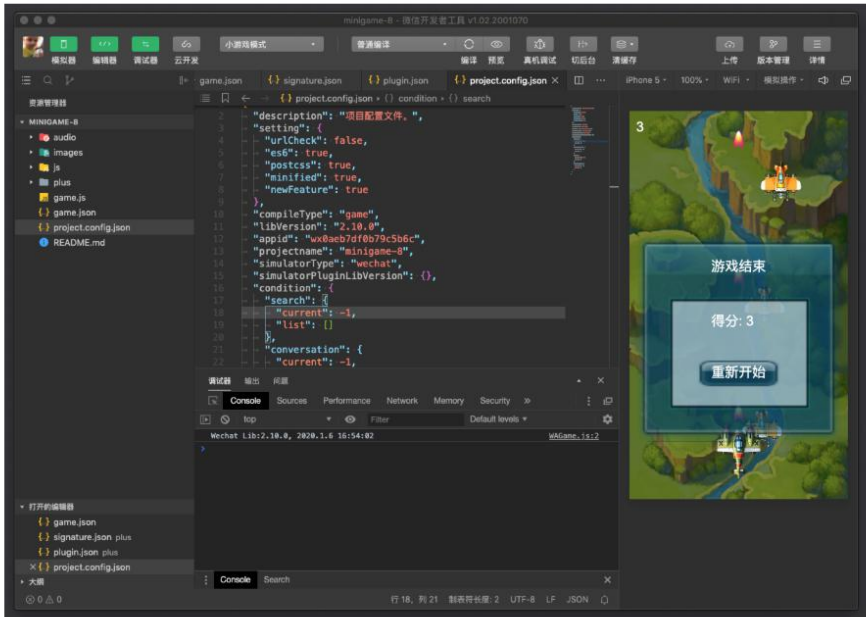
历史更新日志

概览

为了帮助开发者简单和高效地开发和调试微信小程序，我们在原有的公众号网页调试工具的基础上，推出了全新的 微信开发者工具，集成了公众号网页调试和小程序调试两种开发模式。

- 使用公众号网页调试，开发者可以调试微信网页授权和微信JS-SDK 详情
- 使用小程序调试，开发者可以完成小程序的 API 和页面的开发调试、代码查看和编辑、小程序预览和发布等功能。

为了更好的开发体验，我们从视觉、交互、性能等方面对开发者工具进行升级，推出了 1.0.0 版本。



Windows 仅支持 Windows 7 及以上版本。

稳定版 Stable Build (1.02.2004020)

测试版缺陷收敛后转为稳定版；

Windows 64、Windows 32、macOS

预发布版 RC Build (1.02.2003121)

预发布版，包含大的特性；通过内部测试，稳定性尚可

Windows 64、Windows 32、macOS

开发版 Nightly Build

日常构建版本，用于尽快修复缺陷和敏捷上线小的特性；开发自验证，稳定性欠佳

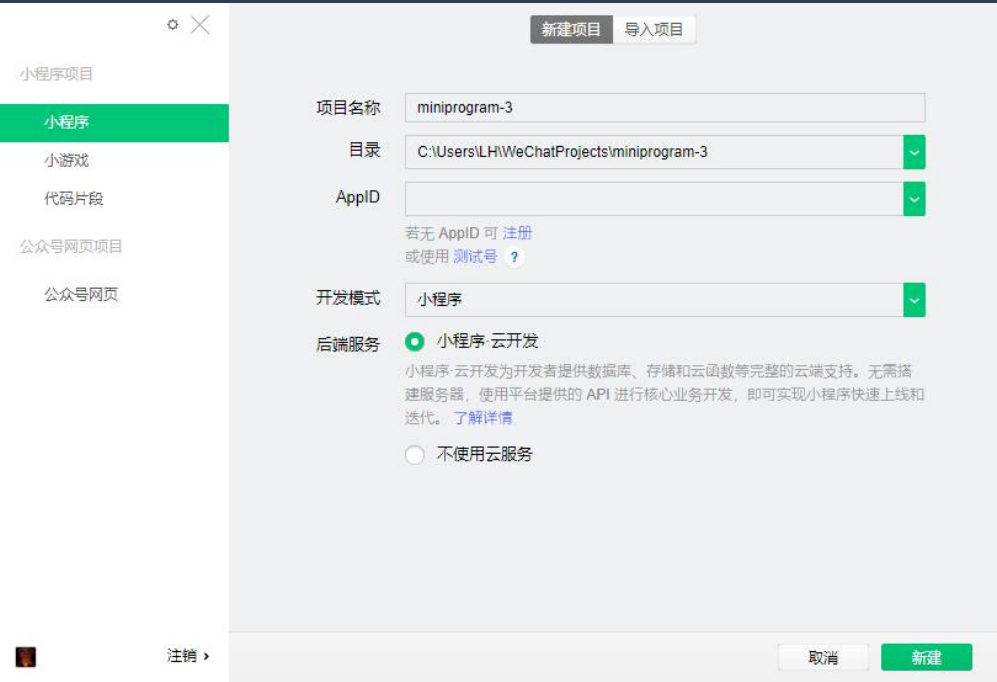
Windows 64、Windows 32、macOS

全部更新日志

全部更新日志

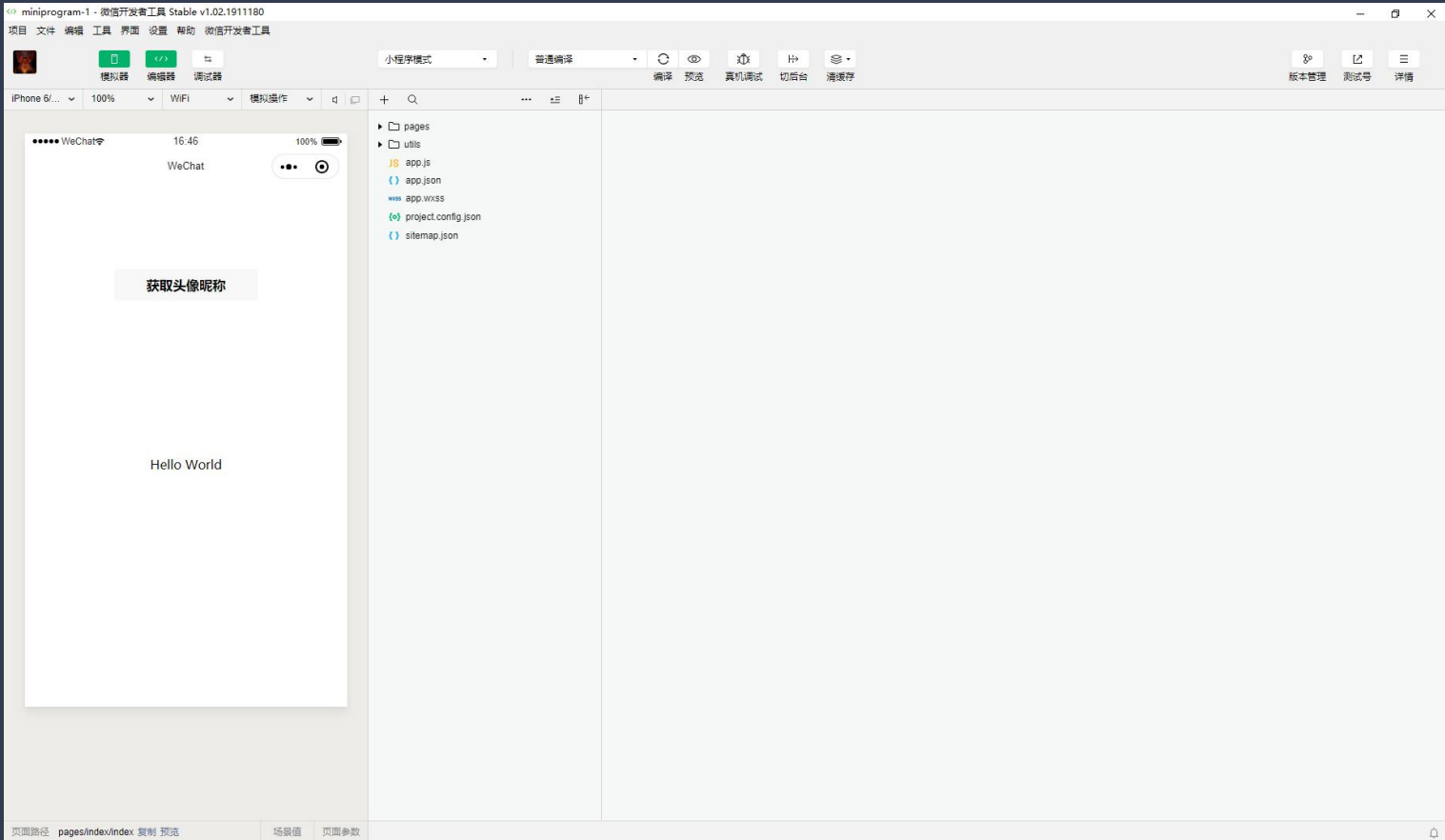
在工具概览界面点击“微信开发者工具”蓝字链接即可进入右侧下载界面，建议选择稳定版下载即可。

微信扫码登录后在右侧新建项目，名称目录自定，AppID可以使用注册后获得的ID，也可以点击测试号进行简单地尝试。



那么，申请小程序以及获取开发者工具我们已经完成了。

新建项目在页面加载完成之后，你就可以见到如下的开发界面了，默认会打开编辑器界面（右）和模拟器界面（左）。



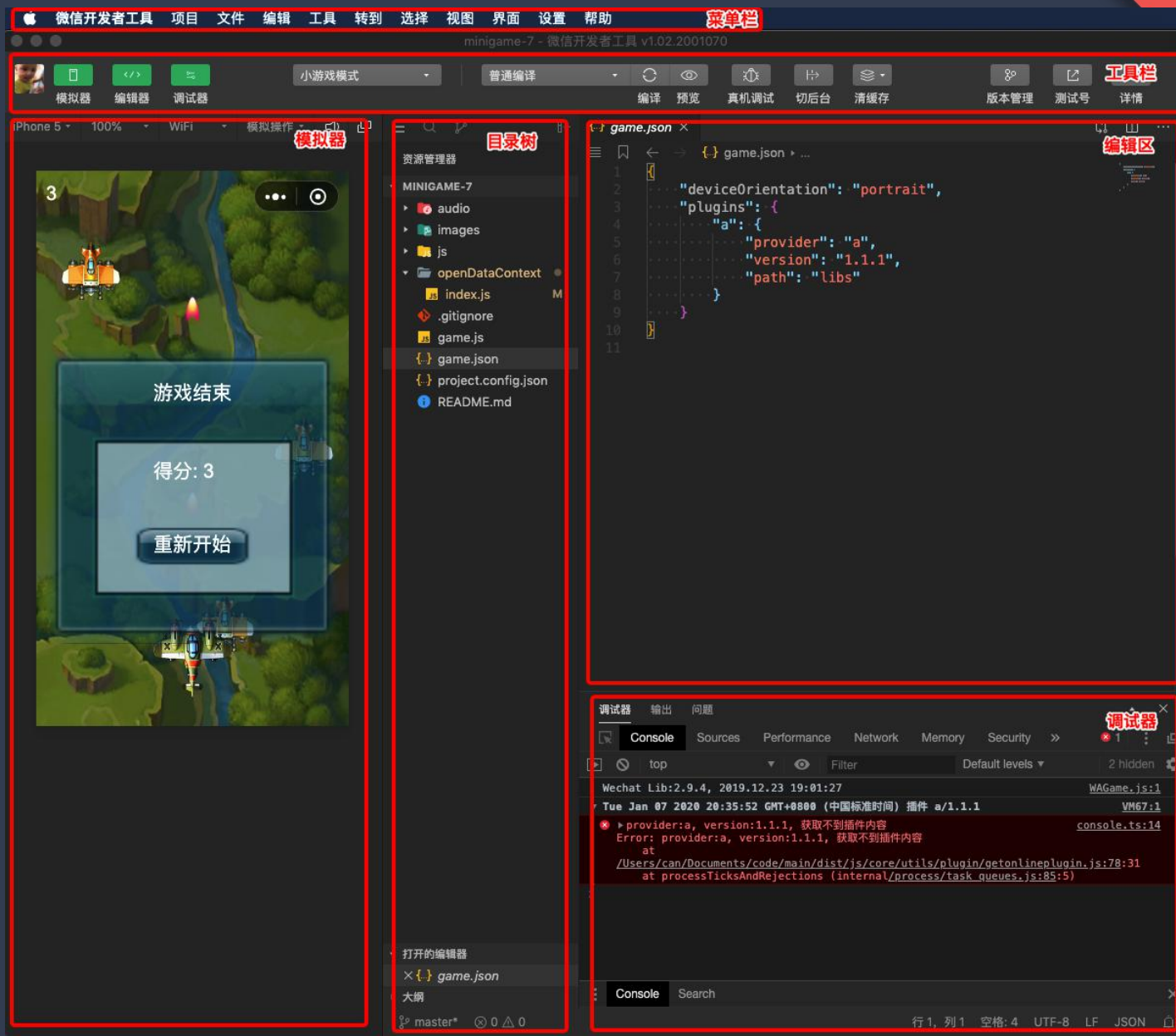
第三部分

开发者工具和开发文档

开发者工具

全新的微信开发者工具，使用小程序调试，开发者可以完成小程序的 API 和页面的开发调试、代码查看和编辑、小程序预览和发布等功能。

目录和编辑区构成常见的编辑器结构，调试器里有常见的控制台和信息反馈，模拟器可以及时对页面编译并进行显示，可以实时进行调整和测试。



开发文档

微信官方给你的“小程序百科全书”。里面有对小程序基本的说明，以及小程序框架和组件的内容，还有大量API的实例。他和开发者工具连接紧密，开发者工具可以及时链接到开发文档获取资料和说明，开发文档也有直接传递数据到开发者工具进行实践的链接。用好开发者工具和开发文档，你的小程序就算是完成入门了！



示例代码

在开发者工具中预览效果

JAVASCRIPT WXML WXSS

```
var types = ['default', 'primary', 'warn']
var pageObject = {
  data: {
    defaultSize: 'default',
    primarySize: 'default',
    warnSize: 'default',
    disabled: false,
    plain: false,
    loading: false
  },
  setDisabled: function (e) {
    this.setData({
      disabled: !this.data.disabled
    })
  },
  setPlain: function (e) {
    this.setData({
      plain: !this.data.plain
    })
  },
  setLoading: function (e) {
    this.setData({
      loading: !this.data.loading
    })
  },
  onGotUserInfo: function (e) {
    console.log(e.detail.userInfo)
  }
}
```



谢谢大家！

你以为真的结束了？

理论讲解是结束了，实践才刚刚开始！