目录

[一、注册小程序开发账号 1](#_Toc525829308)

[1. 注册地址 1](#_Toc525829309)

[2. 选择账号类型 1](#_Toc525829310)

[3. 绑定邮箱 1](#_Toc525829311)

[4. 信息登记 2](#_Toc525829312)

[5. 填写管理员信息 2](#_Toc525829313)

[6. 登录小程序管理平台 3](#_Toc525829314)

[7. 获取AppID 3](#_Toc525829315)

[8. 小程序信息设置 4](#_Toc525829316)

[9. 服务器域名设置 4](#_Toc525829317)

[二、开发IDE 5](#_Toc525829318)

[1. 开发工具 5](#_Toc525829319)

[2. 下载地址 5](#_Toc525829320)

[3. 创建项目 5](#_Toc525829321)

[4. IDE功能 8](#_Toc525829322)

[5. 编译 8](#_Toc525829323)

[6. 上传代码生成体验版 10](#_Toc525829324)

[三、项目结构 10](#_Toc525829325)

[1. 目录结构 10](#_Toc525829326)

[2. 全局文件 11](#_Toc525829327)

[3.页面文件目录 14](#_Toc525829328)

[四、语法（重点） 15](#_Toc525829329)

[1. wxml与html的标签差异对比 15](#_Toc525829330)

[2. wxss文件中无法获取本地资源 15](#_Toc525829331)

[3. 标签绑定点击事件 16](#_Toc525829332)

[4. 数据绑定（重点） 16](#_Toc525829333)

[5. 局部变量的获取与赋值 17](#_Toc525829334)

[6. 全局变量的获取与赋值 18](#_Toc525829335)

[7. 获取用户输入 18](#_Toc525829336)

[8. tab栏设置 18](#_Toc525829337)

[9. 页面间跳转 19](#_Toc525829338)

[10. 小程序及页面生命周期 19](#_Toc525829339)

[五、限制 21](#_Toc525829340)

[六、发布 21](#_Toc525829341)

[七、补充：Flex布局介绍（h5也适用） 23](#_Toc525829342)

[1.flex容器属性 23](#_Toc525829343)

[2.flex元素属性 24](#_Toc525829344)

[3.相对定位和绝对定位 24](#_Toc525829345)

[4.tip: 布局和尺寸单位 24](#_Toc525829346)

[八、工作原理 25](#_Toc525829347)

# 

（配合导航窗格浏览更佳）

# 一、注册小程序开发账号

1. 注册地址

[**https://mp.weixin.qq.com/**](https://mp.weixin.qq.com/)

2. 选择账号类型

小程序



3. 绑定邮箱

**（每个邮箱仅能申请一个小程序，建议不要使用个人邮箱）**



4. 信息登记

**（企业登记步骤会复杂一些）**



5. 填写管理员信息

**（十分重要，后续管理都需要管理员扫码）**



6. 登录小程序管理平台

地址： <https://mp.weixin.qq.com/>

在用户身份目录下可添加开发者、小程序体验者，分配权限

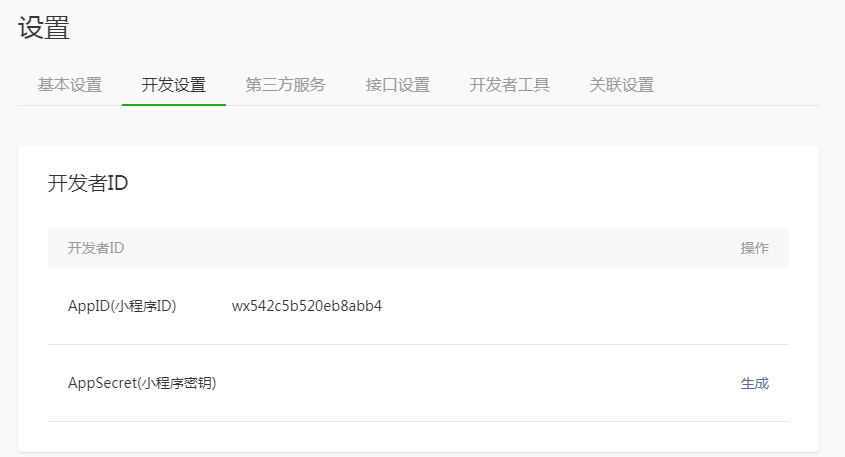
设置开发者后，代码可以上传到微信公众平台，便于协作开发



7. 获取AppID

**（每个小程序使用一个AppID 唯一）**

在设置目录下的开发设置可以获取开发者ID，新建项目时需要



8. 小程序信息设置

在设置页面--->基本设置中，可以设置小程序的各种信息

小程序名称： 设置后一年内可修改两次 （名称不能与其他小程序重复）

小程序头像： 一个月可修改五次

小程序介绍文字： 用户在微信中搜索时显示介绍文字，一个月可修改5次



9. 服务器域名设置

**小程序访问的所有链接都必须在设置页面--->开发设置中声明，否则访问会报错。**

**所有链接必须使用https，且不能使用ip地址访问**



# 二、开发IDE

1. 开发工具

微信web开发者工具

2. 下载地址

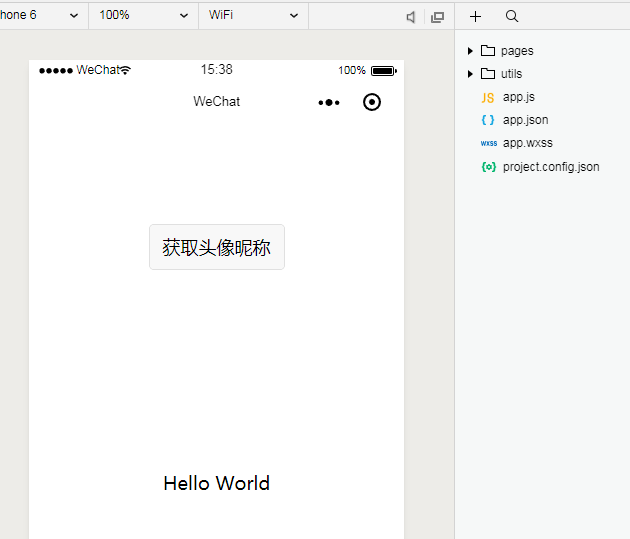
<https://developers.weixin.qq.com/miniprogram/dev/devtools/download.html>

3. 创建项目

写测试demo时可点击下方的测试号AppID，开发正式项目的话需要在管理平台拿到注册的AppID。



选中建立快速启动模板的话，会生成项目会自带绑定微信账号功能，可以获取微信个人信息， 不选择则建立一个空项目



IDE右上角的详情页面，可以设置不校验合法域名，这样可以使用http的域名（方便测试）



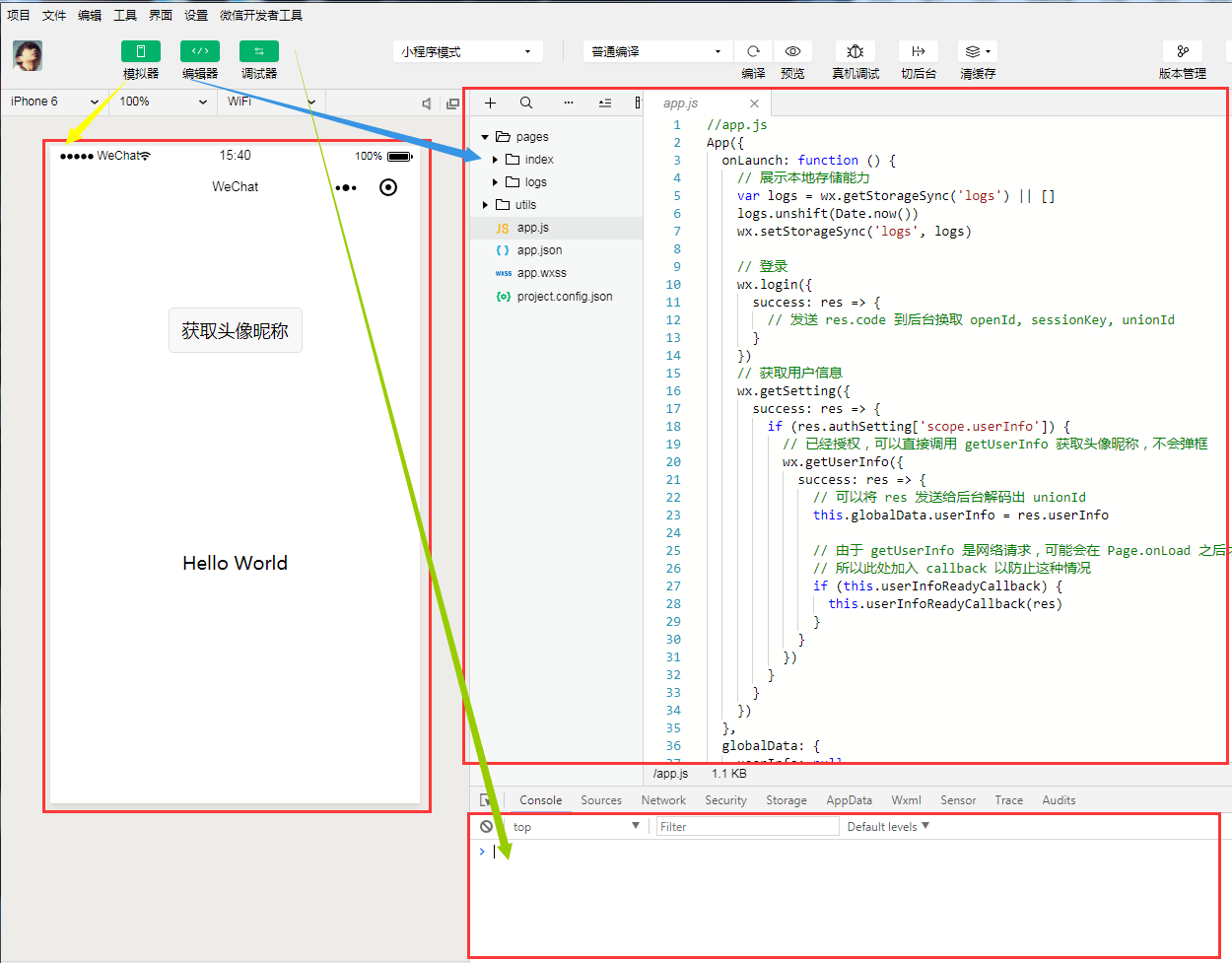
4. IDE功能

主要分为三大视窗，可自定义快捷键调整

模拟器：用于实时查看显示效果

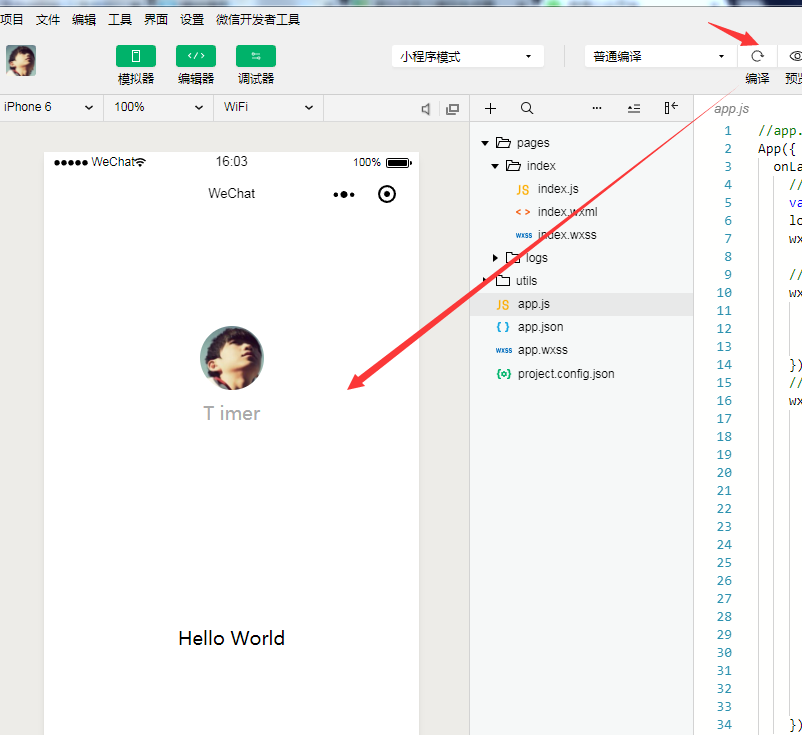
编辑器：代码编辑区

调试器：查看后台调试输出

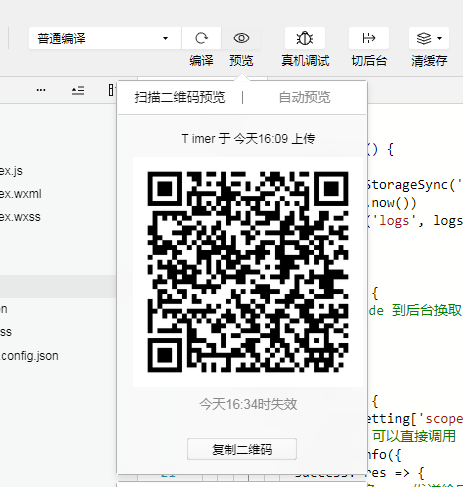


5. 编译

点击编译按钮是IDE编译，可在左侧模拟器实时看到程序显示效果

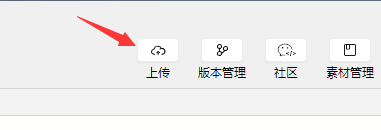


点击预览按钮则会生成二维码，使用微信扫描二维码可在手机上测试（项目未发布前，需要管理平台-🡪用户身份中添加了测试人员 设置体验者权限）



6. 上传代码生成体验版

（体验人员同样需要权限）

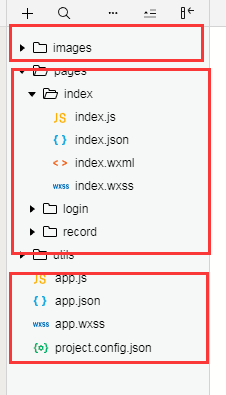




# 三、项目结构

1. 目录结构

小程序的目录可分为三部分： 资源文件夹、项目页面文件夹、项目配置文件。



项目中所有的页面都放在pages文件夹下

新建页面： 在pages文件上右键--🡪新建-🡪目录

然后在新建的目录上右击-🡪新建-🡪page 即可自动生成四个文件

.json后缀的JSON配置文件

.wxml后缀的WXML模板文件

.wxss后缀的WXSS样式文件

.js后缀的JS脚本逻辑文件

2. 全局文件

app.json、 app.js 、app.wxss是小程序的全局文件

app.json文件是当前小程序的全局配置，包括了小程序的所有页面路径（新建页面时会自动在app.json文件中添加）、界面表现、网络超时时间、底部tab等。

详请链接：

https://developers.weixin.qq.com/miniprogram/dev/framework/config.html#%E5%85%A8%E5%B1%80%E9%85%8D%E7%BD%AE



page字段：用于描述当前小程序所有页面路径 （排列于第一个的表示默认启动页面）

window字段：定义小程序所有页面的顶部背景颜色，文字颜色定义等。

tabBar字段 （重点）：如果小程序是一个多tab应用（客户端窗口的底部或顶部有tab栏可以切换页面），可以通过tabBar配置项指定tab栏的表现，以及tab切换时显示的对应页面。



networkTimeout字段：各类网络请求的超时时间，单位均为毫秒。

debug字段：在开发者工具的控制台面板，调试信息以info的形式给出，可以清晰的查看项目的启动过程。

app.js文件 可以编写小程序的生命周期函数以及存放全局变量、全局函数等。



可以自定义全局字段

3.页面文件目录

新建页面： 在pages文件上右键-🡪新建-🡪目录

然后在新建的目录上右击-🡪新建-🡪page 即可自动生成四个文件



index.js为脚本逻辑文件

index.json为JSON配置文件，用于配置单个页面的样式风格

index.wxml为模板文件，充当网页编程中的类似HTML的角色

index.wxss为样式文件，类似于CSS，但是有些许不同

# 四、语法（重点）

（主要介绍与h5不同之处）

1. wxml与html的标签差异对比

<view> 标签 相当于html中的 div、body等，即小程序中都是通过使用view的嵌套来实现的。

<text> == <span> text标签是唯一一个可以长按选中文本的标签，而且text标签不允许互相嵌套，还支持转义符“/”

<navigator> == <a> navigator标签目前只能作为站内导航，不支持链接外跳。

<image> == <img> image标签是图片标签，它并非真正意义上的图片标签，更像是给一个<view>设置背景图片，它提供了mode属性来满足各种背景设置需求，也就是文档中的12中模式，其中包括3中缩放模式、9中剪裁模式。

<input> == input[type=’text’]

<checkbox> == input[type=’checkbox’]

<radio> == input[type=’radio’] 以上三种标签其实是拆分了HTML中的input标签。

2. wxss文件中无法获取本地资源

如果想在wxss文件中为某个标签设置背景，只能使用网络url链接，否则会报错。

 background-image：url( ../images/bg.png  ); 错误

解决：在wxml中使用image标签作为背景，图片设置在属性中。

或者 将图片转换为base64格式的。也可以将背景图放在服务器中。

3. 标签绑定点击事件

小程序中如果想给某个标签添加点击事件，只需在标签的属性中添加bind属性即可，属性的内容为js中的函数

**wxml文件**

<view class='btn\_loginout'>

<button bindtap='loginout'>退出登录</button>

</view>

**js文件中写函数**

loginout: function () {

wx.showModal({

title: '提示',

content: '是否确认退出登录？',

success: function(res) {

if (res.confirm) {

wx.setStorageSync('isLogin', false)

wx.redirectTo({

url: '../index/index',

})

}

}

})

},

4. 数据绑定（重点）

为了项目方便管理与结构清晰，小程序增加了类似MVVM的开发模式，提倡把渲染和逻辑分离，简单来说就是不要再让JS直接操控DOM，JS只需要管理状态即可，然后再通过一种模板语法来描述状态和界面结构的关系即可。

通过{{ }}的语法把一个变量绑定到界面上，称为数据绑定。仅仅通过数据绑定还不够完整的描述状态和界面的关系，还需要if/else, for等控制能力，在小程序中，这些控制能力都用wx: 开头的属性来表达。

**1）绑定数据：**

wxml文件：

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

Js文件中设置变量message的值

Page({

data: {

message: 'Hello MINA!'

}

})

**设置组件属性** （用数据绑定设置属性时，必须使用双引号，不能用单引号）

<view id="item-{{id}}"> </view>

Page({

data: {

id: 0

}

})

**三元运算**

<view hidden="{{flag ? true : false}}"> Hidden </view>

**2）列表渲染wx:for**

在组件上使用wx：for控制属性绑定一个数组，即可使用数组中各项的数据重复渲染该组件。

默认数组的当前项的下标变量名默认为index，数组当前项的变量名默认为item。

<!--wxml-->

<view wx:for="{{array}}"> {{item}} </view> //会渲染出5个view

// page.js

Page({

data: {

array: [1, 2, 3, 4, 5]

}

})

**3）条件渲染**

在框架中，使用wx:if=”{{condition}}”来判断是否需要渲染该代码块。

<view wx:if="{{condition}}"> True </view>

也可以使用wx:elif 和 wx:else来添加一个else块 (elif为else if的简写)

<view wx:if="{{length > 5}}"> 1 </view>

<view wx:elif="{{length > 2}}"> 2 </view>

<view wx:else> 3 </view>

5. 局部变量的获取与赋值

1）局部变量声明

需要存储的变量一般都存放在当前页面的data中（即js文件中）

Page({

/\*\*

\* 页面的初始数据

\*/

data: {

flag: 1,

adcode: []

},

})

2）获取变量的值

this.data.flag

3）更改变量的值

this.setData({

flag: 2

})

6. 全局变量的获取与赋值

1）变量的声明（app.js文件中）

App（{

globalData:{

userphone: ‘’,

permission: ‘’

}

}）

2）获取变量的值

首先，获取app的实例

var app = getApp() //放在单个页面js文件的开头部分

取值: app.globalData.userphone

3）更改变量的值

app.globalData.userphone = ‘xxxxx’

7. 获取用户输入

1）wxml文件中绑定输入检测函数

<input placeholder='请输入工作票号' bindinput='ticketinput'/>

2）js文件中实时获取

ticketinput: function (e) {

this.setData({

ticketinput: e.detail.value //实时赋值

})

},

输入框每次修改时都会调用

8. tab栏设置

在app.json中设置tabBar属性，tabBar栏最多只能设置5个，最少两个。

"tabBar": {

"color": "#56C5C9", //默认文字颜色

"selectedColor": "#ffffff", //选中时文字颜色

"backgroundColor": "#0A777C", //tab栏背景颜色

"list": [

{

"pagePath": "pages/main/main", //所导航的页面路径

"text": "首页", //tab栏文字

"iconPath": "images/address\_normal.png", //默认显示图片

"selectedIconPath": "images/address\_checked.png" //选中时显示的图片

},

{

"pagePath": "pages/record/record",

"text": "记录",

"iconPath": "images/record\_normal.png",

"selectedIconPath": "images/record\_checked.png"

}

]

}

9. 页面间跳转

1）tab页面跳转到非tab页面

wx.redirectTo({

url: '../index/index', //页面在栈中pop，返回键无效

})

2）非tab页面跳转到tab页面

wx.switchTab({

url: '../main/main', //页面在栈中pop，返回键无效

})

3）非tab页面之间的跳转

wx.navigateTo({

url: '../index/index ', //页面仍保留在栈中，可以按返回键返回上一个页面

})

wx.redirectTo({

url: '../index/index', //页面在栈中pop，返回键无效

})

10. 小程序及页面生命周期

**小程序的生命周期**

App({

/\*\*

\* 当小程序初始化完成时，会触发 onLaunch（全局只触发一次）

\*/

onLaunch: function() {

},

/\*\*

\* 当小程序启动，或从后台进入前台显示，会触发 onShow

\*/

onShow: function(options) {

},

/\*\*

\* 当小程序从前台进入后台，会触发 onHide

\*/

onHide: function() {

},

/\*\*

\* 当小程序发生脚本错误，或者 api 调用失败时，会触发 onError 并带上错误信息

\*/

onError: function(msg) {

},

globalData: {

userphone: '', //用户手机

code: '', //用户编码 表明权限区域

permission: '', //用户权限

wechat: ''

}

})

**单个页面的生命周期**

Page({

/\*\*

\* 页面的初始数据

\*/

data: {

},

/\*\*

\* 生命周期函数--监听页面加载

\*/

onLoad: function (options) {

},

/\*\*

\* 生命周期函数--监听页面初次渲染完成

\*/

onReady: function () {

},

/\*\*

\* 生命周期函数--监听页面显示

\*/

onShow: function () {

},

/\*\*

\* 生命周期函数--监听页面隐藏

\*/

onHide: function () {

},

/\*\*

\* 生命周期函数--监听页面卸载

\*/

onUnload: function () {

},

/\*\*

\* 页面相关事件处理函数--监听用户下拉动作

\*/

onPullDownRefresh: function () {

},

/\*\*

\* 页面上拉触底事件的处理函数

\*/

onReachBottom: function () {

},

/\*\*

\* 用户点击右上角分享

\*/

onShareAppMessage: function () {

}

})

# 五、限制

1. 小程序项目代码包大小限制为2m

解决办法：

项目的本地资源图片尽量小

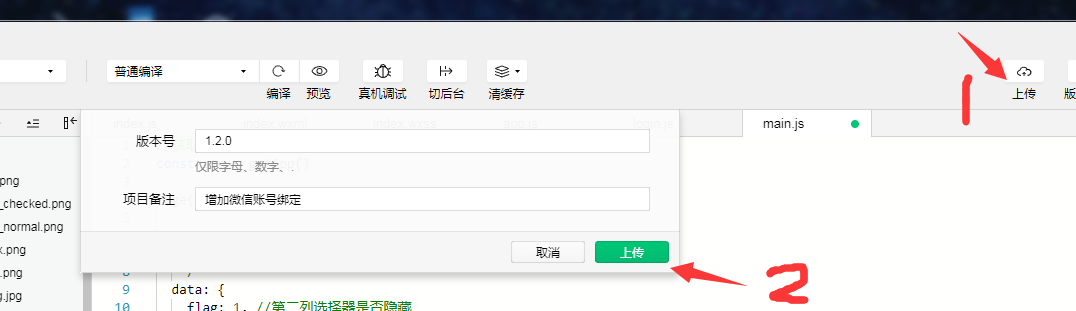
分成多个包加载，但每个包不超过2M，总大小不超过4M。

2. getLocation方法在定位（即获取用户经纬度坐标时），移动网络与wifi会存在差异，有时会导致获取不到坐标

解决办法：个人暂未找到

# 六、发布

1. 首先点击上传，将项目上传到管理平台作为体验版



2. 在开发管理平台可以看到刚刚上传的开发版本（即体验版），点击提交审核，如有登录功能，则需要提供测试账号，以便审核人员测试。



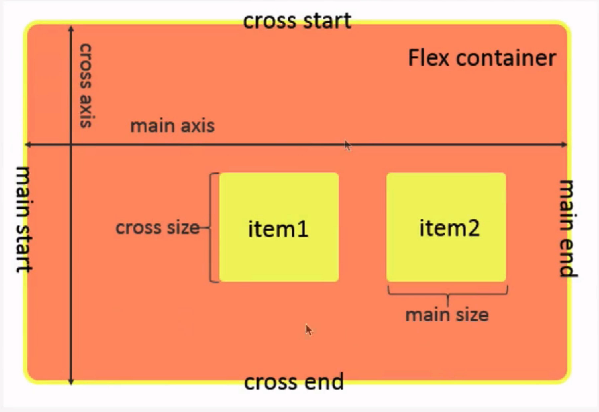
3. 点击提交审核后会看到审核版本，第一审核时间会久一点（2-3天），后面相对会快一点。



4. 审核通过后（会发送通知信息到微信）即可发布。

5. 小程序发布之后，第一次打开后会自动下载更新（但是不会应用），这时候可以将小程序的后台杀死（或者切到后台，5分钟后小程序的进程会自动销毁），第二次打开即为最新版本。

# 七、补充：Flex布局介绍（h5也适用）



main axis为主轴 cross为交叉轴

**使用：**wxss添加属性 display： flex；

1.flex容器属性

**flex-direction** 决定元素的排列方向

row 横向 (左到右为主轴) column 竖向 (上到下为主轴)

**flex-warp** 决定元素如何换行

nowarp 不换行 warp 换行 warp-reverse 反向换行 第一行在最下面

**flex-flow** flex-direction和flex-wrap的简写

**justify-content** 元素在主轴上的对齐方式

center 居中 flex-start、 flex-end、 space-around 间距相等(两端有空隙)

space-between 间距相等(两端贴边)

**align-items** 元素在交叉轴的对齐方式

.../ stretch(当子组件没有设置高度时 自动占全屏) baseline元素以文字对齐

2.flex元素属性

**flex-grow** 当有多余空间时，元素的放大比例 0为不放大 1为均分

**flex-shrink** 当空间不足时，元素的缩小比例 0不缩小

**flex-basis** 元素在主轴上占据的空间 ：250rpx;

**flex** 以上三个的简写

**order** 定义元素的排列顺序 order：2(将当前元素排在第二位)

**align-self** 定义元素自身的对齐方式 center 。。。

3.相对定位和绝对定位

相对定位的元素是相对自身进行定位，参照物是自己。

绝对定位的元素是相对离它最近的一个已定位的父级元素进行定位。

4.tip: 布局和尺寸单位

在布局时尽量使用百分比布局

width: 100%;

height: 100vh; //使用100%获取不到

元素大小尽量使用rpx

rpx是微信小程序解决自适应屏幕尺寸的尺寸单位，微信小程序规定屏幕的宽度为750rpx，小程序会按照实际的手机的屏幕宽度来进行换算。

如：iphone6屏幕宽度为375px， 所以iphone6中 1rpx = 0.5px

微信小程序也支持rem单位，rem规定屏幕的宽度为20rem，

所以1 rem = （750 / 20）rpx = 37.5 rpx

# 八、工作原理

小程序的视图层目前使用 WebView 作为渲染载体，而逻辑层是由独立的 JavascriptCore 作为运行环境。在架构上，WebView 和 JavascriptCore 都是独立的模块，并不具备数据直接共享的通道。当前，视图层和逻辑层的数据传输，实际上通过两边提供的 evaluateJavascript 所实现。即用户传输的数据，需要将其转换为字符串形式传递，同时把转换后的数据内容拼接成一份 JS 脚本，再通过执行 JS 脚本的形式传递到两边独立环境。而 evaluateJavascript 的执行会受很多方面的影响，数据到达视图层并不是实时的。