

小程序开发入门

我折腾小程序也有段时间了，公司也开发了多款工具类小程序，本 Chat 很高兴能和各位小程序开发者一起交流学习，主要分享小程序入门讲解，包括一些我在开发中踩过的坑。希望通过本篇文章的分享，能让更多的小程序入门者少走弯路。

如何注册一个小程序

第一种方式，直接去官网注册。

官方注册地址：<https://mp.weixin.qq.com/>，点击右上角的“立即注册”。



然后选择“小程序”：

请选择注册的帐号类型



接下来，填写小程序的注册信息，这里我就不详细说了，根据自己的需要选择小程序类型是“个人”，或者“企业”（企业用户需要填写认证信息）：

GitChat

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

邮箱

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

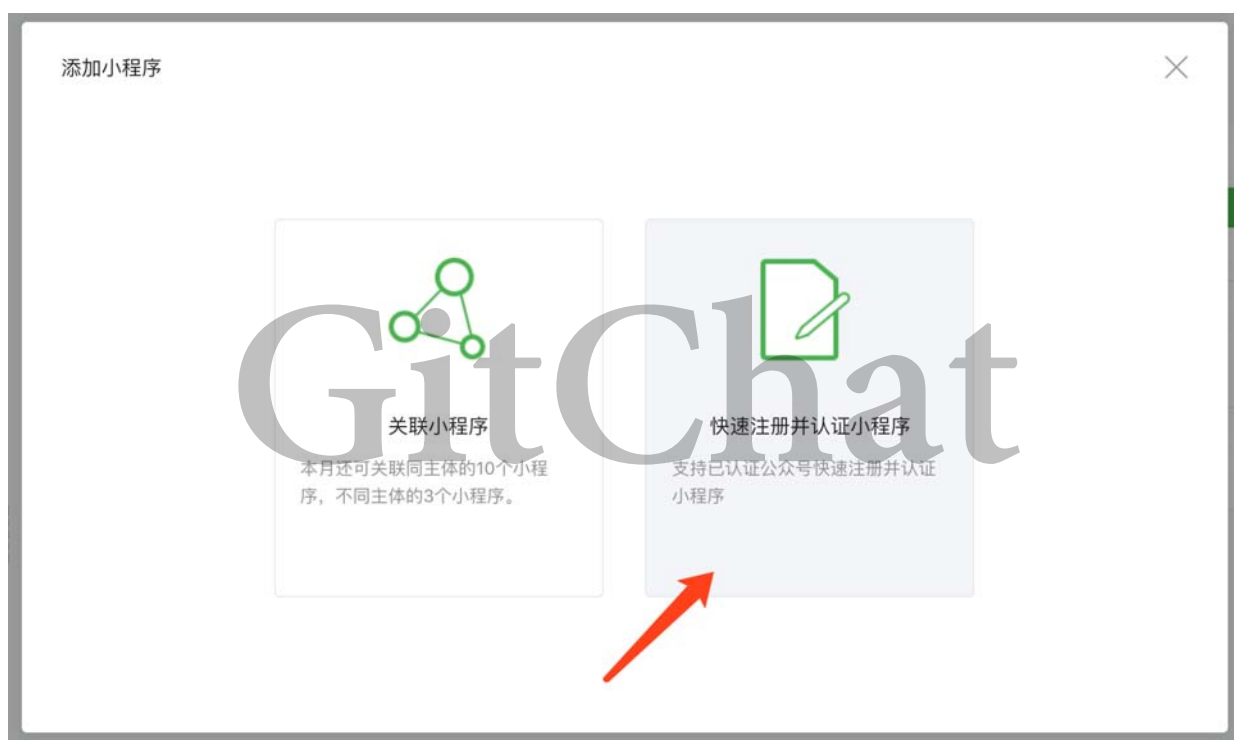


换一张

填写完信息之后，直接提交审核，微信团队会有个审核过程，然后给你反馈审核结果。

第二种方式，去公众号后台，小程序管理里面注册。

其步骤包括：登录已有公众号后台 -> 小程序管理 -> 添加 -> 快速注册并认证小程序。



这种方式注册的小程序，如果你的公众号已经认证过了，小程序的认证是可以复用公众号的认证的。微信的认证是按次收费，所以，用这种方式注册，可以省下一部分费用支付。

注意：个人类型的小程序，有部分权限是没有的，比如获取微信用户绑定的手机号码，微信支付功能，微信卡券功能都必须是企业类型的小程序才有权限。

并且小程序目前也没办法更换主体，因此，建议大家注册之前小程序类型不要选择错了。

小程序有哪些宣传方式

主要有以下几种宣传方式：

1. 小程序搜索入口 & 附近的小程序；
2. 扫一扫、长按识别小程序码；
3. 好友、群分享的小程序卡片；
4. 已关联小程序的公众号菜单或文章；
5. 第三方小程序应用商店；
6. 小程序之间互相跳转；
7. 群聊中发送过的小程序卡片。

以上几种方式是小程序主流的宣传方式，后续还可能会有新的形式出现，比如，近期小程序又出了个“功能直达”的内测，部分优秀的小程序已经获得了内测资格，其中我参与开发的小程序，有四款已经获取内测资格。

功能直达样式如下（感觉百度的直达号看到这功能后，会哭晕）：



小程序开发工具讲解

工欲善其事，必先利其器。微信官方提供了专门的小程序开发工具，即“微信开发者工具”，经过不断的版本更新，此工具目前已经比2017年刚发布时，好用多了。

点击[这里](#)，访问[下载地址](#)。

我来简单介绍一下微信开发者工具。

首先进行下载，安装这里就不说了，安装完成之后，直接打开，会出现如下界面：



扫码之后会出现小程序 & 公众号网页开发两个选项：



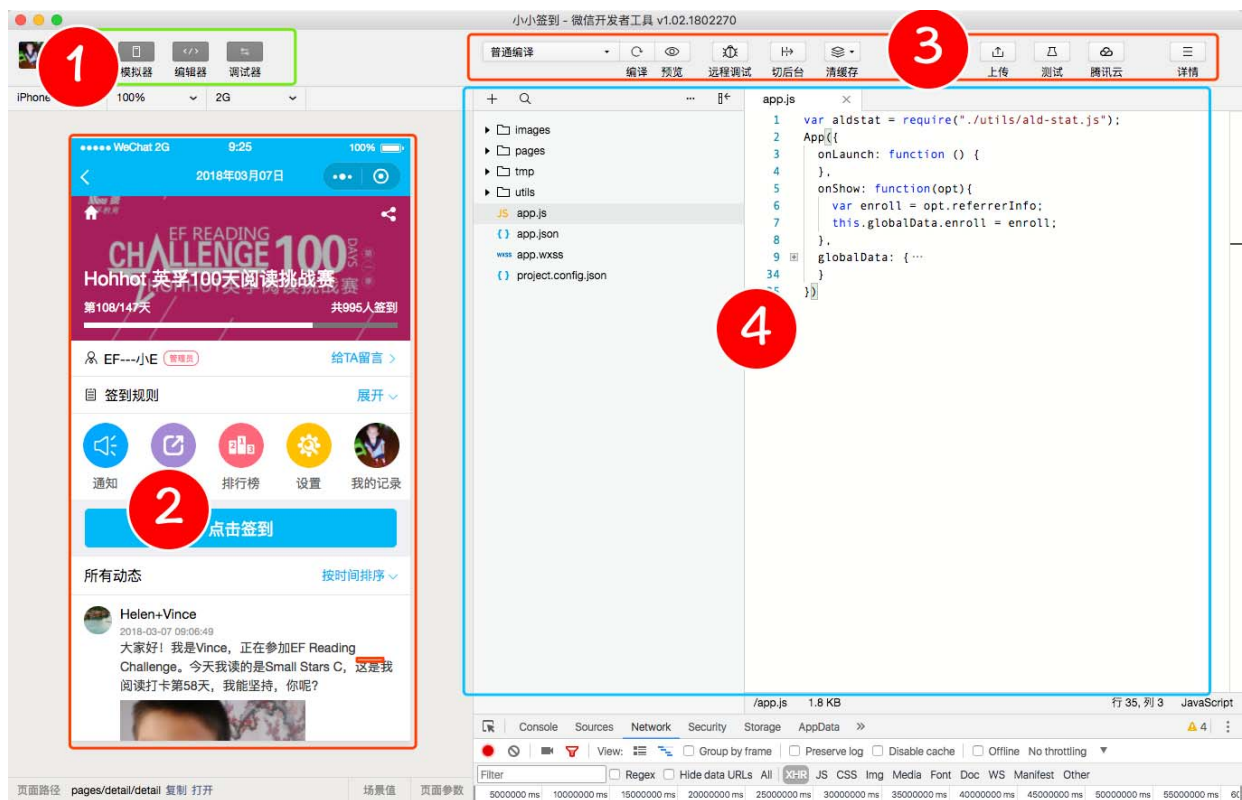
微信开发者工具

v1.02.1802270

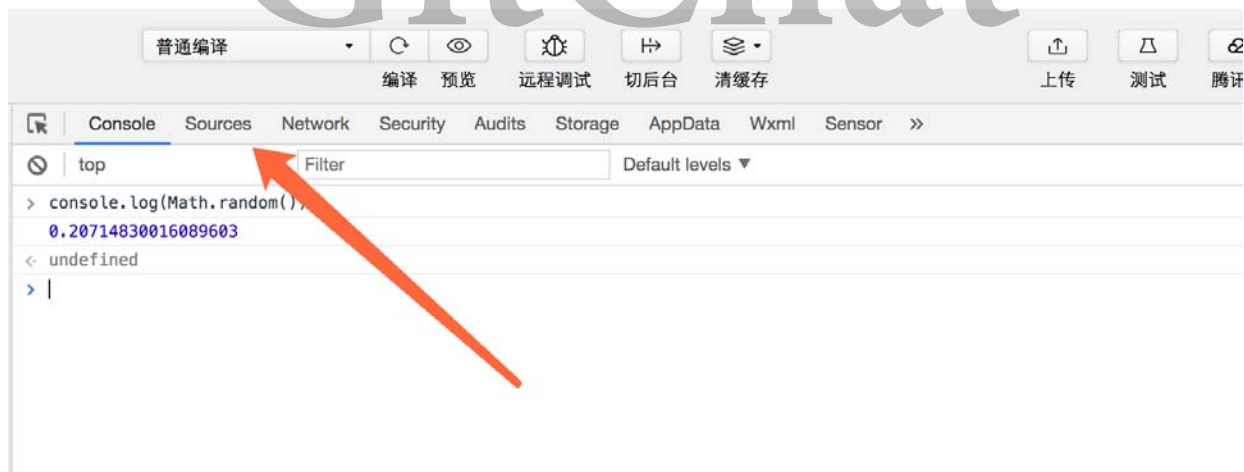


切换帐号 >

也就是说，微信开发者工具，是可以用来开发小程序、公众号网页以及小游戏的一款工具。这里我们直接选择小程序，然后我打开了一个现有的项目：



- 图中①区域主要是切换②和④区域时用到，我这里定义为“功能切换区”；
- 图中②区域是“模拟器区域”，这里是对小程序的预览展示效果；
- 图中③区域是一些常用操作，我给定义为“属性操作区域”；
- 图中④区域是“编辑器区域”，分为左边“代码结构” & 右边“代码编写区”。这里可以通过①区切换为“调试器”，如下：



我这里要特别说明一下，③区域最右侧的“详情”按钮，点开之后，如下图所示：



下方的“项目设置”，有四个复选框，新入门的同学，请一定不要勾选最下方的“不校验安全域名”这个选项。因为在我的小程序开发群里，已经有多位同学在这儿踩过坑，表现为，体验版本的小程序数据都正常展示，上线后发现所有数据都不展示了。这就是因为域名没有做 TLS/SSL 认证，关于小程序入门的一些注意点儿，欢迎查看这篇文章：[微信小程序开发-常见问题](#)。

编写一个简单小程序

1.WXSS 单位建议用 rpx。

官方推荐单位用 rpx，默认屏幕宽度是 750rpx，以 iPhone6 手机为基准进行设计的。

$750 = 375 * 2$ ，然后其他屏幕尺寸时，小程序会自动进行缩放。

之前做过移动端开发的同学肯定也用过 640px 的设计稿， $640 = 320 * 2$ ，320px 是 iPhone4、4S、5、5S 的实际网页宽度，375px 是 iPhone6、6S 的实际网页宽度。移动端适配方案这里就不详细说了，在小程序内只需要用 rpx 做单位即可。

2.wxml 组件只能用官方组件。

官方组件地址，请访问[这里](#)。

小程序中，不能使用 div、p、ul、h1 这样的标签，只能使用官方组件，并且所有组件都必须成对存在，或者单闭合形式存在，否则会报错。

编写代码部分，这里不详细说，大家可以查看我写的一个 Demo：



微信扫一扫，使用小程序

小程序的代码提交、设置体验版、发布

代码提交，从开发工具中提交即可，如下图所示：



设置体验版本和发布上线都需要登录小程序后台，地址为：<https://mp.weixin.qq.com>。

和公众号共用同一个地址，用户名和密码是注册小程序时填写的，登录后台之后，有这几个地方需要注意。

1.把某个开发者的代码设置为体验版本。



2.添加体验者权限。



3.提交审核、发布上线、撤回等操作。



4.告警群设置。



小程序后台可以查看所有错误信息，但是，为了方便第一时间了解错误报警，建议使用官方“客户端告警群”，如上图。告警的阈值，可以自己设置。

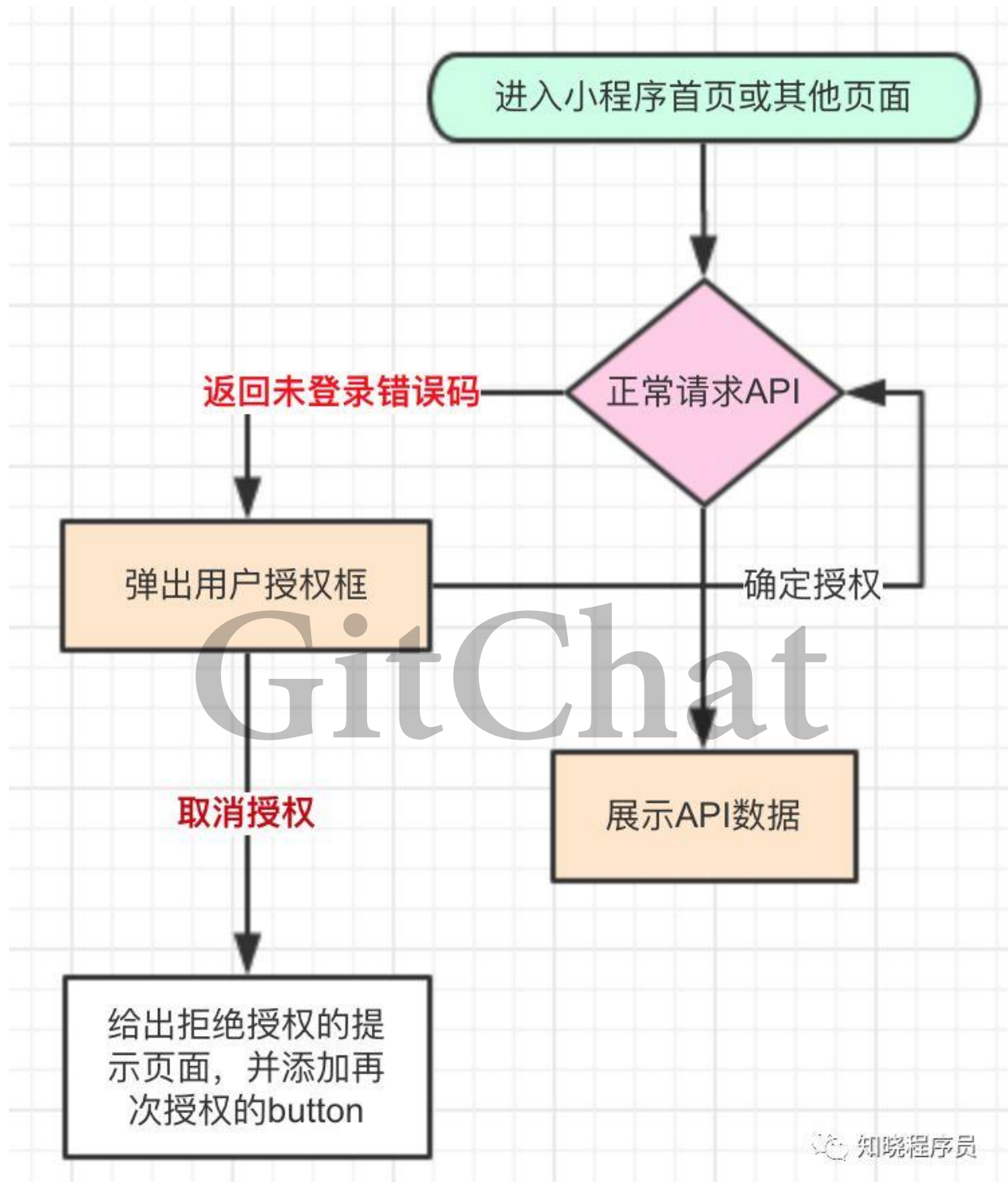
小程序开发常见问题讲解

以上讲的都是皮毛，为了让大家更多的了解小程序该如何做，我这里列出几个常见的技术问题，希望对大家有帮助。

1.小程序登录逻辑的实现。

小程序登录的实现，官方建议自己保存用户登录状态，不要频繁调用 `wx.login`，否则会限制登录。

这里说一下我自己实现登录的逻辑，无代码，实现逻辑看下图：



2. 小程序返回首页的实现。

这里说个返回首页的场景，比如首页可以点击 button 进入详情页（`wx.navigateTo`），详情页此时再回首页，可以直接 `wx.navigateBack` 即可。

如果详情页面允许分享，用户就可能直接通过分享出去的小程序卡片进入详情页，此时，回首页的逻辑就与上面说的不同了，必须再重新打开首页。详细请看下方代码。

```
function toIndex() {
  var pageCount = getCurrentPages().length;
  if (pageCount > 1) {
    wx.navigateBack({
      delta: 1
    })
  } else {
    wx.redirectTo({
      url: '../index/index'
    })
  }
}
```

页面互相跳转的逻辑，尽量设计简单一些，否则的话，很坑人。比如，A 页可以跳至 B 页面，B 页面又可以跳至 C 页面，B 和 C 页面都允许分享，此时，从 C 页面返回 B 页面的逻辑就比上面的要复杂。

3.防止用户多次点击。

比如用户提交表单数据，点击 submit 按钮，需要调用保存数据的 API，如果不做误点击处理，用户可能会多次点击 submit，这样就会保存冗余数据。

可以提取公用方法到 util.js 中，如下：

```
function buttonClicked(self) {
  self.setData({
    buttonClicked: true
  })
  setTimeout(function () {
    self.setData({
      buttonClicked: false
    })
  }, 500)
}
```

知晓程序员

WXML：

```
<form report-submit="true" bindsubmit="create">
  <view class="...">
```

知晓程序员

JavaScript：

```
create: function (e) {
  if (this.data.buttonClicked) return;
  util.buttonClicked(this);
```

知晓程序员

500毫秒以内的点击都只会处理一次，时间长短自己可以调整。

4.小程序之间如何传值。

(1) URL 传值

这种方式最常用，比如：

```
wx.navigateTo({
  url: '../detail/detail?id='+id+'&access_token='+access_token
})
```

这里面直接通过跳转页面的 URL 进行传值，然后在另一个页面进行接收：

```
onLoad: function (opt) {
  console.log('cid =' + opt.cid);
  console.log('access_token =' + opt.access_token);
}
```

这种传值方式只适合值比较少的时候使用，传值比较多时候，还是建议写本地缓存。

(2) 本地缓存

小程序 API 提供了本地缓存数据的 API，默认可以缓存10M的数据，如下：

```
wx.setStorageSync('testData', testData);
```

testData 可以是一个 object，在需要的页面直接调用 wx.getStorageSync 即可获取，这样就解决了传值较少的问题了。

(3) 全局 App

其实还有第三种方式，就是全局 App 变量。app.js 和 app.wxss 中的代码都是全局生效的，所以我们可以利用这一点儿，在不同页面之间进行传值。

```
App({
  onLaunch: function () {

  },
  globalData: {
    host: 'https://xxx.com',
  }
})
```


也可以在其他 JavaScript 里面动态修改 globalData，如 `getApp().globalData.host = 'XXX';`。

5. 获取小程序表单数据。

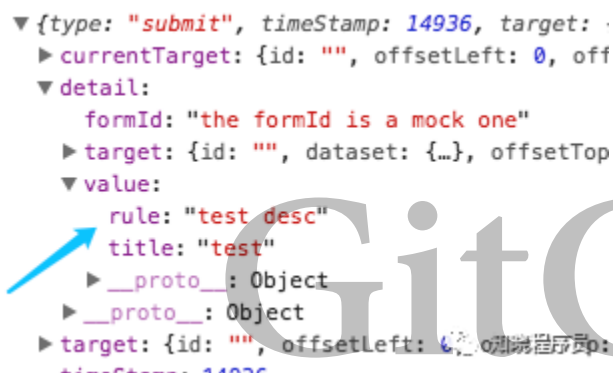
做过小程序的同学，都知道小程序中是通过数据渲染页面的，没办法获取 Dom 节点，表单提交就不能像 H5 页面那样去获取表单项的见容了。

小程序中的表单提交必须用户手动触发，不能通过 JavaScript 自动提交。

获取表单数据有两种方式。

(1) 获取 event 中的值。

正常的 form 表单提交，都可以在 `event.detail.value` 中获取到页面表单项填写的值，如下：

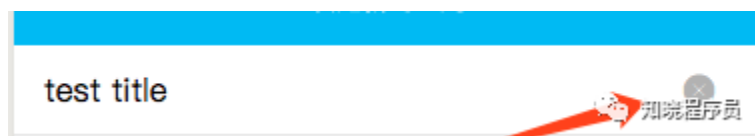


```
{type: "submit", timeStamp: 14936, target: {
  currentTarget: {id: "", offsetLeft: 0, off
  detail: {
    formId: "the formId is a mock one"
    target: {id: "", dataset: {...}, offsetTop
    value: {
      rule: "test desc"
      title: "test"
      __proto__: Object
    }
    target: {id: "", offsetLeft: 0, offsetTop: 14936
```

这里需要在 WXML 中的，把 input、textarea、radio 等表单项设置 name 属性，上图中的 title，就是 input 的 name 属性。

```
<input type="text" name="title" auto-focus='true' />
```

这种方式获取表单数据很方便，但是，如果需要对表单数据有清除功能，我们该如何实现呢？如下图所示。



我们先来看看第二种方式。

(2) 通过设置变量值保存表单数据。

这种方式也比较好实现，就是给表单项绑定事件，当内容发生变化时，set 一下变量值。所以，提交表单的时候直接获取变量值就 OK 了。


```
<input type="text" bindinput="inputTitle" name="title" value="{{title}}" auto-focus='true' />
```

可以给 input 绑定事件 (bindinput="inputTitle")，然后在 inputTitle 里面简单处理一下：

```
inputTitle: function (e) {  
    this.setData({  
        title: e.detail.value,  
        titleEmpty: e.detail.value.length == 0  
    })  
},
```

上面的 titleEmpty 是为了判断 title 是否为空，如果为空，就不显示右侧的“清除 icon”。这种方式很容易实现上面说的清空内容。

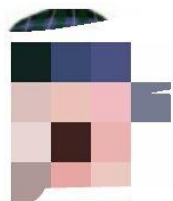
在 form 的 submit 时，直接 `var title = this.data.title;` 就获取到了表单数据，很方便。

6. 如何获取更多的 formId

相信使用过小程序的同学，多少都收到过小程序的通过消息，如下：

GitChat

微信



昨天



昨天



服务通知

投票结果通知

昨天



昨天

GitChat

知晓程序员



服务通知



抽奖提醒 安佩国 发起的抽奖已开奖，点击
查看中奖名单（3 天未达指定 50
人，按 5 人开奖）

进入小程序查看



昨天 12:05



投一个



投票结果通知

1月16日 12:05

投票标题 2018年汕尾市城区捷胜文昌中学全校绘画美术作品比赛

投票类型 多选，匿名

投票发起人 蔡威俊

投票结果 第一名是「高中部 高三 刘海川 素描」，共1229票，占比14.0%，点击查看完整结果...

参与人数 8053

进入小程序查看

GitChat



知晓程序员

这类通知消息，是和好友消息一样展示在微信的聊天列表中，所以，点击率还是比较高的。想实现这种小程序的模板消息，就必须获取用户的 formid 才可以（如何发消息，请仔细查阅[小程序官方文档](#)）。

我们来说一下如何获取 formid：

1. 必须通过 form 组件提交才能获取到 formid；
2. 给 form 组件设置 report-submit="true" 属性；
3. 给 form 组件添加 bindsuit 事件绑定，携带 form 中的数据触发 submit 事件，
event.detail = {value: {'name': 'value'}, formId: ''}；
4. 必须用户手动触发提交表单，不能 JavaScript 模拟提交，所以，页面上必须要有提交按钮。

看一下示例代码：

```
<form report-submit='true' bindsuit='userSubmit'>
  <button class='button' bindtap='copy' form-type='submit'>复制
</button>
</form>
```

以上示例就可以在 userSubmit 里获取到 formId 了：

```
userSubmit: function (e) {  
  console.log(e.detail.formId);  
},
```

需要注意一点，开发工具里面是没办法查看到真实的 formId 的，会是这样一句提示：“the formId is a mock one”，提交给服务端就可以拿到了。

7. 小程序页面跳转后的刷新逻辑

场景是这样，用户从列表 A 页面跳至 B 页面，并在 B 页面做了更新操作（编辑或删除信息），当用户再 back 到 A 页面时，需要对 A 页面的数据进行刷新；如果用户未做更新操作，back 回 A 页面后，就不需要对 A 页面数据刷新。

我们可以在 B 页面操作时，写入一个数据在本地，如下代码：

```
formSubmit: function(){  
  // 操作之后，本地缓存一个值  
  util.setStorage('needRefresh', true);  
},
```

返回 A 页面时，再到 page 的 onShow 方法中去判断一下就 OK。

```
onShow: function(){  
  var needRefresh = util.getStorage('needRefresh');  
  if(needRefresh){  
    // 这里写入你的刷新逻辑  
    .....  
    util.removeStorage('needRefresh');  
  }  
},
```

8. 小程序如何判断转发到群还是好友

这个问题，要区分一下分享之前和分享之后就好办了。

当小程序分享之前，小程序无法区分到底是分享给了好友，还是分享给了微信群。也就是说，你想在分享之前去做些事情，这是不可以的，做不到。

分享之后，有间接的办法来知道小程序卡片分享的是好友还是群，先看一下截图：



(重要通知小程序)

我们今天主要来说，分享后是如何核对好友卡片和群卡片的。实现思路：通过场景值来判断。

1001: 发现栏小程序主入口
1005: 顶部搜索框的搜索结果页
1006: 发现栏小程序主入口搜索框的搜索结果页
1007: 单人聊天会话中的小程序消息卡片
1008: 群聊会话中的小程序消息卡片
1011: 扫描二维码
1012: 长按图片识别二维码

上图是小程序官方给的场景值，已经对单人聊天会话和群聊天会话做了区分。

我们可以在 app.js 里面的 onLaunch 方法或者 onShow 方法获取到场景值：

```
App({
  onLaunch: function (opt) {
    console.log(opt.scene);
  },
  onShow: function (opt) {
    console.log(opt.scene);
  }
});
```

拿到值之后，你可以保存全局变量，或者写入本地缓存，在相应的 page.js 里面去做判断，如果是群聊，显示群聊界面，如果是私聊，显示私聊界面。

上面只说了如何区分群聊还是私聊，那么仅限某个群可见这功能又如何实现呢？

思路是：分享小程序卡片之后，获取到群 id，当群用户点击群内小程序卡片时，也会获取一个群 id，这两个群 id 相同，则表明是同一个群；没有获取到群 id，表明不是从群聊中打开小程序卡片；两个群 id 不同，则表明不是同一个群。

上面已经说了，有两种场景可以获取群 id。

第一，小程序卡片分享到群聊中。分享完成之后，有个分享成功的回调方法，可以在此方法中获取到 shareTickets：

```
onShareAppMessage: function () {
  var self = this;
  return {
    title: '您收到了一条重要通知',
    path: '/pages/...',
    success: function (res) {
      console.log(res.shareTickets[0]);
    },
    fail: function (res) {
      console.log('转发到群/好友失败 =====');
    }
  }
},
```

拿到 shareTickets 之后，可以调用 wx.getShareInfo，获取解密 gid 会用到的向量和加密值，传给服务端进行解密（小程序中所有的解密操作，都放在服务端进行）。

注：上面成功回调时，res.shareTickets 是个 list，因为分享给好友和群的时候，可以多选，最多选9项。

第二，从群聊中打开小程序卡片时。从群中打开小程序卡片，会在 app.js 里面的 onLaunch 方法和 onShow 方法中获取到 shareTickets。

```
onShow: function (ops) {
  this.shareTicket = '';
  if (ops && ops.scene && ops.scene == 1044) {
    this.shareTicket = ops.shareTicket ? ops.shareTicket : '';
  }
},
```

拿到 shareTickets 之后，去服务端解密。

相信大家也有看到过，有些小程序做了群排行榜功能。实现方式，也是同样的道理。

9. 小程序之间互相跳转的注意点儿


小程序之间互相跳转，使用 wx.navigateToMiniProgram 来实现，从[这里](#)访问官方文档。

wx.navigateToMiniProgram(OBJECT)

基础库 1.3.0 开始支持，低版本需做兼容处理

iOS 微信客户端 6.5.9 版本开始支持，Android 客户端即将在 6.5.10 版本开始支持，请先使用 iOS 客户端进行调试

打开同一公众号下关联的另一个小程序。

 知晓程序员

上面有一句话，要注意：**打开同一公众号下关联的另一个小程序。如果没有关联同一个公众号，则无法成功打开另一小程序。**

(1) 公众号关联小程序。

公众号关联小程序后，将可在图文消息、自定义菜单、模板消息等功能中使用小程序。

关联规则如下：

所有公众号都可以关联小程序。

1. 一个公众号可关联10个同主体的小程序，3个不同主体的小程序。
2. 一个小程序可关联3个公众号。
3. 公众号一个月可新增关联小程序13次，小程序一个月可新增关联5次。

关联流程如下：

登录公众号后台 -> 小程序 -> 小程序管理 -> 添加 -> 关联小程序。

如果你已经关联成功，那就可以看一下代码的实现了，如下：

```
if (wx.navigateToMiniProgram) {  
  wx.navigateToMiniProgram({  
    appId: recommend.appid,
```



```


        path: '/pages/detail/detail?cid=' + cid
    })
} else {
    wx.previewImage({
        urls: [recommend.qrcode],
    })
}
}

```

wx.navigateToMiniProgram，从基础库 1.3.0 开始支持，低版本需做兼容处理。低版本时，我们就直接给用户显示一张带二维码的图片，告知用户需要升级微信版本，或者扫描二维码才能进入。

测试的时候，需要跳到另一个小程序的体验版，可以设置 envVersion 参数：develop（开发版），trial（体验版），release（正式版）。

		要打开的小程序版本，有效值 develop（开发版），trial（体验版），release（正式版），仅在当前
envVersion	String 否	小程序为开发版或体验版时此参数有效；如果当前小程序是体验版或正式版，则打开的小程序必定是正式版。默认值 release

 知晓程序员

如果你公司的小程序产品比较多，又想把数据打通，这种方式是个不错的选择~

10.公用方法建议写在 util.js 中

开发小程序过程中，大家都可能有自己的提取公用代码逻辑，我建议大家把公用方法放在 util.js 中。

app.js 中主要获取 options 参数时用，比如识别小程序码，需要获取 scene 参数；从群聊中打开小程序卡片，你可以获取 shareTickets 参数；小程序之间互相跳转，你可以获取到 appid 等参数。

部分 util.js 方法如下：

```

function postFormId(formId) {
    wx.request({
        url: getApp().globalData.host + '/v1/formid',
        data: { 'formid': formId, 'access_token': getToken() },
        method: 'POST'
    })
}

```



```
function toIndex() {  
  var pageCount = getCurrentPages().length;  
  if (pageCount > 1) {  
    wx.navigateBack({  
      delta: 1  
    })  
  } else {  
    wx.redirectTo({  
      url: '../index/index'  
    })  
  }  
}
```

```
function buttonClicked(self) {  
  self.setData({  
    buttonClicked: true  
  })  
  setTimeout(function () {  
    self.setData({  
      buttonClicked: false  
    })  
  }, 500)  
}
```

```
function getStorage(key) {  
  try {  
    wx.getStorageSync(key);  
  }  
  catch (e) {  
    getStorage(key);  
  }  
}
```

小程序开发过程中，问题肯定不止这几个，实在是太多，这里只是列出几个常见问题，欢迎各位和我一起来讨论技术问题，或者交流一下在小程序开发过程中踩过的坑。

本次文章分享到此结束，希望对大家入门小程序有帮助。文章，如有错误的地方，希望各位同学帮忙指正，如果是交流小程序开发的技术问题，也可以添加我微信：mianhuabingbei。