



minapp
.com

Powered by
ifanr //

微信小程序 入门指南



知晓程序 著

重新定义 App,
小程序如何
影响未来?

目 录

关于作者.....	3
前言.....	5
第一节 什么是小程序?	7
第二节 小程序将给我们带来什么改变	12
第三节 小程序上手案例	18
知了交通.....	19
剁手吗.....	24
知了地铁.....	32
解忧室.....	36
第四节 如何转行小程序开发?	41
第五节 如何拥有自己的小程序?	45
如何注册微信小程序	46
如何搞定微信认证?	52
如何完善小程序信息?	59
开发者工具怎么用?	65
小程序的审核与发布	74
第六节 小程序官方文档解读	83
开发文档解读	84
设计文档解读	94
运营文档解读	104
第七节 电商类小程序开发教程	112
如何做轮播图	113
如何做商品列表	123
数据加载和图文排版	134
写在最后	148
附录.....	149
小程序大事记	149

关于作者

爱范儿是首批获得微信小程序内测资格的 200 个公司之一，也是全球第一个开发出微信小程序的公司。

知晓程序，是爱范儿旗下专注小程序生态的新品牌。

我们提供最全面、新鲜的小程序资讯、观点、教程、Demo、活动等内容和服务，在这里你能了解到关于小程序的一切。

我们还跟全国各地的小程序关注者开展了深入交流，形成了面向小程序开发者的未来小程序·黑客马拉松 / 面向非技术从业人员（产品 / 运营 / 市场等）的未来小程序·workshop / 面向小程序爱好者的未来小程序·MindTalk 的活动矩阵。

目前，知晓程序共包含四大版块：

- 知晓程序微信公众号：小程序资讯媒体
- 知晓程序商店：小程序应用商店；
- 知晓程序社区：小程序交流平台；
- 知晓程序 BaaS：小程序后端服务平台。

你可以通过微信公众号「知晓程序」和网站 minapp.com 了解最新进展。

前言

微信小程序（应用号）正式公布后，在业内引起了轩然大波。

爱范儿主编刘学文曾经在《腾讯微信自己造了个 App Store，自此它的目标是苹果和 Google》提及：

朋友圈把普通用户拉进来，做成了社交网络；公众号把内容创作者拉进来，做成了最大的内容平台；现在小程序开始尝试拉拢开发者，做成一个森林式的服务平台。而这一切，都是在一个应用内完成，而且还是一个目前看起来逻辑层级还算明晰的应用内。

回想起苹果的发展历程，人们每每记得的是 iPhone 初代的发布，iPhone 4 的惊艳亮相，但是很少人会说起 App Store 的出现，也正是苹果 App Store 的出现，才成就了苹果的闭环有了生态。现在小程序相对于微信的意义，某种程度上类似于 App Store 之于苹果的意义。

在小程序拨开云雾始见青天的过程中，仍然有很多人并不清楚：什么是小程序？会对我们的生活、工作有什么影响？对互联网行业、移动生态格局意味着什么？

面对新机遇和新时代，这并不是最好的姿态。所以，才有了这本书的诞生。

本书收录了微信小程序正式发布前（注：目前处于公测阶段）的所有关键信息。你不仅会知道小程序是什么、会带来什么变化、真正用起来的感觉如何，还能收获一份简易又不失细节的入门教程，包括如何拥有自己的小程序、小程序开发的部分注意事项等等。

简单来说，它能让你成为最懂小程序的人之一。

我们还试图通过本书传达一种态度：机会留给有准备的人。

iPhone 刚刚面世的时候，许多人不以为意；App 开发热潮开始的时候，许多人没有兴趣；微信公众号席卷全民的时候，许多人冷眼旁观……

历史总在发生，你想让自己错过吗？

第一节 什么是小程序？

小程序到底是什么？它跟 app 和 HTML 5 有什么区别？它和订阅号、服务号又有什么不同？小程序真的是无所不能的吗？

现在，知晓程序（微信号 zxcx0101）就用这篇文章，为你揭开小程序的面纱。

一、 小程序到底是什么？

「微信之父」张小龙是这样描述小程序的：

小程序是一个不需要下载安装就可使用的应用，它实现了应用触手可及的梦想，用户扫一扫或者搜一下即可打开应用。也体现了用完即走的理念，用户不用关心是否安装太多应用的问题。应用将无处不在，随时可用，但又无需安装卸载。

简单来说，小程序不用安装就能使用；它的体积也非常小，每一个都不超过 1M。

小程序的到来，将给我们带来许多便利和好处：

1. 少了安装 App 的麻烦

在 app 时代，如果我们要使用一项功能，通常需要完成以下步骤：

- 在 App Store 或 Android 应用市场，寻找能实现相应功能的应用；
- 下载并安装该应用；

- 在手机桌面，找到应用；
- 打开并使用应用。

而在小程序时代，只需要搜索或者「扫一扫」，即可打开应用。没有了下载安装环节，人和功能的连接变得更加简单直接。

2. 释放手机内存

除了使用更加便捷之外，小程序的「小体积」，也能为大家的手机大大减轻负担。

相信大家手机里都装了不少 app。一般情况下，安装一个 app，得花上几十 MB、上百 MB，甚至几 GB 的内存。手机容量小的人，一旦多装了几个 app，就要面临内存不够的困扰。

在这种情况下，大小不超过 1 M 的小程序，就成了手机内存不够用的救星。

而且，小程序还不需要安装，那些平时大部分时间用不上，又不得不装的 app，一旦变身小程序，就可以彻底地从你的手机中移除了。

3. 让手机桌面更简洁

小程序问世后，也许，你手机桌面上的许多 app 将会消失。

那些功能简单、使用频率低的 app 将会被小程序替代，它们不会在手机桌面上再占据一席之地，而是折叠在微信这个超级 app 里面，等到你使用时再「召唤」出来。

目前，微信小程序的入口还没有完全公布，希望小程序帮我们「清理」了手机桌面以后，微信的界面也能一如既往地保持简洁的样貌。

二、 小程序与 HTML 5 有什么区别？

小程序出生以后，有很多人将它与 HTML 5 进行比较，甚至有人把它们混淆在一起。

但实际上，小程序和 HTML 5 本质上是两种不同的东西：小程序是计算机程序，而HTML 5 则是互联网网页。

1. 对用户而言

我们打开 HTML 5，实际上是打开一个网页，而网页需要在浏览器中渲染。

所以，在微信上点开 HTML 5，需要完成网页加载的步骤，这就给人一种「卡」的感觉。

而对于微信小程序来说，它的代码可以直接在微信上运行，省去了通过浏览器渲染的步骤，在体验上会胜过 HTML 5 不少。

总而言之，在微信中使用小程序，会比 HTML 5 流畅很多。

2. 对开发者而言

对于开发者来说，相比 HTML 5，小程序可以节省大量的服务器资源。

那是因为，HTML 5 需要开发者的服务器向用户发送 HTML 5 代码，而微信会帮开发者分发他们的小程序。

三、 小程序与订阅号、服务号有什么区别？

同属于微信的生态体系，小程序也被很多人拿来跟订阅号、服务号比较。

那么，它们之间到底有什么区别呢？我们先看看订阅号和服务号到底是什么：

- **订阅号：**顾名思义，是用户在微信中订阅文章所使用的公众帐户。订阅号可以向用户推送文章和信息，也可以管理订阅用户、与用户交流。它是一种媒体属性的产品。
- **服务号：**是一种服务导向的公众帐户。它可以帮助机构在微信中向用户提供服务，同时，服务号还可以让用户直接与机构的客服系统进行沟通、为用户提供品牌信息等。

以上两者都是以「聊天界面」为基础的，即是说，它们的功能和界面是早就规定好了的。

而小程序允许开发者自定义界面，所以，不用局限于微信的聊天界面。开发者不仅可以提供更好的使用体验，也无需考虑对接公众帐户对话，开发门槛是比较低的。

简单来说，如果把这三个平台比喻成三件玩具，那么，订阅号和服务号可能是公仔或模型车，而小程序则是可以自由拼装的乐高积木。

四、 小程序是「无所不能」的吗？

虽然小程序有千般好，但小程序也并不是万能的。

在做小程序之前，我们还需要清晰地认识到小程序的局限性：

首先，**小程序不能承载所有的用户需求**。无论是游戏娱乐、文档处理等「重需求」，还是带有传播能力的营销需求，小程序都不能很好地满足。如果要满足这些需求，**app** 或 **HTML 5** 会是更好的选择。

其次，小程序的体系整体依赖于微信。如果你的业务与微信设定的规则有冲突，或主营业务在微信渗透力不强的地区（如国外），微信小程序就不是一个好的选择。

另外，小程序目前不能主动发送消息（包括群发和模板消息）。如果你需要向用户提供客服业务，或是希望向用户推送信息，订阅号或服务号等具有相应功能的帐户形态则更加适合。如果想在小程序中新增客服功能，还需要开发者自己去实现。

希望大家看完这篇文章，能对微信小程序有一个立体而理性的认识。

第二节 小程序将给我们带来什么改变

小程序出现后，很多人都在翘首观望。

应该拥抱小程序吗？小程序会带来什么新变化？要如何应对这种改变？

知晓程序（微信号 [zxcx0101](#)）现在从行业、职业和用户三个层面，来谈谈小程序将给我们带来什么影响。

一、对行业的影响

小程序的服务范围包含 17 个行业。如果你所在的行业名列其中，恭喜你自动获得了小程序的入场券。

关注微信号 [zxcx0101](#)，回复「服务范围」，查看小程序的所有适配行业。

但是，这个世界已经有 app、网站和公众号了，小程序又能给你带来什么好处呢？我们分不同情况看一看。

1. 线下门店

每个线下门店，如餐馆、服装店、旅店，不一定能拥有自己的 app 和网站，也不一定需要拥有自己的 app 和网站。

但每个线下门店，都可以拥有自己的小程序：它能实现 app 和网页的功能，且开发成本更低，用户使用起来也更方便。

以餐馆为例，如果你是一家餐馆的老板，估计已经完成了以下「触网」工作：

- **宣传**：加入大众点评，做团购，做微信公众号
- **支付**：接入支付宝和微信支付

但是，在用户进店到吃完买单之间，还是在用老办法：以服务员为中介，完成点菜、送餐、买单整套流程。

如何提高效率和用户体验？自己做一个点餐系统显然是不现实的，但开发一个点餐小程序却很合适。

而且，这个小程序使用起来会十分方便：用户只需坐在餐位上扫描二维码，就能进入小程序直接点餐和支付。过程简单、迅速，服务员的人力成本都省了。

未来线下门店的小程序，也许会像微信支付和支付宝买单一样普遍。

如果你在经营线下门店，那么可以根据自己的业务需求，主动拥抱小程序。

2. 互联网行业

对于现在的互联网企业来说，app 算是一个标配。

然而，app 的开发、维护和用户获取的成本都比较高。而且，在新的 app 或者在 app 的新功能推向市场之前，谁也不能保证它能受到用户欢迎。要是不受欢迎，对公司和程序员们来说，都是极大的损失。

所以，开发成本低的小程序，是一个提前试错和节约成本的好方法。

如果你身处互联网行业，完全可以把新功能做成小程序，投入市场观察反应，之后再考虑是否需要在客户端上实现。

3. 低频刚需的行业领域

如果将一个行业继续细分，可以分出非常多细微的垂直领域。

但若每一个处在低频刚需行业领域的公司都推出一个 **app**，那么很多人的手机存储就要「告急」了。而且这些需求是不确定的，用户不会也不能在这些 **app** 中长时间停留。

以上这些，都让这些公司推出一个 **app** 变成了一种「资源浪费」。

而微信小程序「小」的形态，可以很好地承载垂直细分行业、满足长尾需求。

处在低频刚需行业领域的公司，可以以更低的成本提供体验更好的小程序；而用户也不再担心「使用一次服务就要装一个 **app**」的情况了，他们只需要扫一下二维码，就能在以后轻松使用这些服务。

二、对职业的影响

1. 前端开发者

在小程序推出之后，很多人发现，「小程序的语法与前端语言很像」，进而给出了「这是前端开发的春天」的结论。

的确，对于前端开发者来说，从网页前端迁移至小程序开发几乎零成本。不仅如此，小程序的 **MINA** 框架可以让前端开发在同等的开发时间里，开发出体验更棒的网页。

但也因为 **MINA** 的限制，让原本在 **web** 领域很好用的开发框架无法继续使用。**Web** 前端程序员可以选择自己实现组件效果，或是使用微信提供的视觉组件框架。

当然，这种限制也会催生一种东西：专用于微信小程序的开发框架。

但更重要的是，在可预见的未来，各家企业对小程序的需求，会令前端程序员的需求出现持续增长。

那么，前端程序员需要做哪些准备呢？

如果前端人员需要向微信小程序发展，除了原有的 HTML、CSS 和 JavaScript，你现在就应该开始了解和学习除这三个以外的前端开发框架。

一个好消息就是，微信已经将小程序的接口封装得很好，前端开发者在开发的时候，只需要简单几行代码，就能轻松调用它们。

2. 后端开发者

「前端开发的春天」，只是指对前端的技术难度降低，并且前端开发可以直接触及最热门的小程序开发而已。

但「简约而不简单」的小程序，对后端的要求一点也没有降低。

对于大部分的小程序来说，它们一样都有数据交换的需求，这就需要后端的支持了。

可以说，小程序的火热并没有「摧毁」后端，反而给后端提出了更高的要求。

小程序的极低开发成本，让一个很小的需求也能通过小程序承载。未来也势必会有公司瞄准这一市场，为企业后端提供支持服务，甚至将后端封装成可独立安装的整包出售。

所以，后端开发者并不需要太过紧张，小程序的出现，只会让后端的需求变得更多。当然，如果你有兴趣，也可以学习并不困难的小程序开发，没准，这就是你从单纯的后端转变为全栈开发者的起点。

3. 个人开发者

虽然，小程序暂时没有开放个人的注册资格，但知晓程序推测，在不久的将来，个人也能像注册公众帐户一样注册小程序。

小程序的学习成本和开发成本降低，所有人都有机会接触小程序编程，也可以通过小程序做出一个像样的东西出来。

与此同时，小程序的分发和使用全部依赖微信，注册小程序帐户就能分发小程序。而微信是一个跨平台操作系统，开发者不再需要担心多平台适配问题。

另外，小程序的低开发成本也会让更多的个人开发者接受小程序的外包工作。开发难度更低的小程序让小程序开发的成本降低，进一步降低外包成本。

而它所带来的影响，就是在鼓励更多的人学习编程、鼓励更多的人将想法转变为实际行动。

在未来，也许小程序会像订阅号一样，成为中国个人开发者的首选选项之一。

三、 对用户的影响

如果你只是一个既不会写代码，又不用操心企业经营的吃瓜群众。那小程序会给你带来什么影响呢？

那就是，你的手机将变得更加好用、更加强大：不用装 **app**，也能实现很多平时所需的功能。那些偶尔才会用到 **app**，可以彻底从手机消失了。

在使用小程序的时候，完全可以对它召之即来，挥之即去，想用就用，用完就走。

可惜的是，这也会给你带来一点副作用：你将会前所未有地依赖你的手机，以及微信这个超级 **app**。

好在，微信小程序现在还未正式发布。我们就一起等待那个到处都是小程序的新世界吧。

第三节 小程序上手案例

作为第一批拿到小程序内测资格的企业，爱范儿开发出了几个有趣、有用的小程序。

为什么要开发这些小程序？它们有什么功能？在开发过程中，有什么值得分享的地方？

知晓程序（微信号 [zxcx0101](#)）将逐个跟大家分享。

知了交通

我们要分享的第一款小程序，叫「知了交通」。

一、「知了交通」功能详解

「知了交通」是一款轻量级的交通查询小程序，功能十分简单明确。

首页有「选择城市」和「输入查询信息」两个功能区。点击当前城市，可进入搜索页面更换城市。

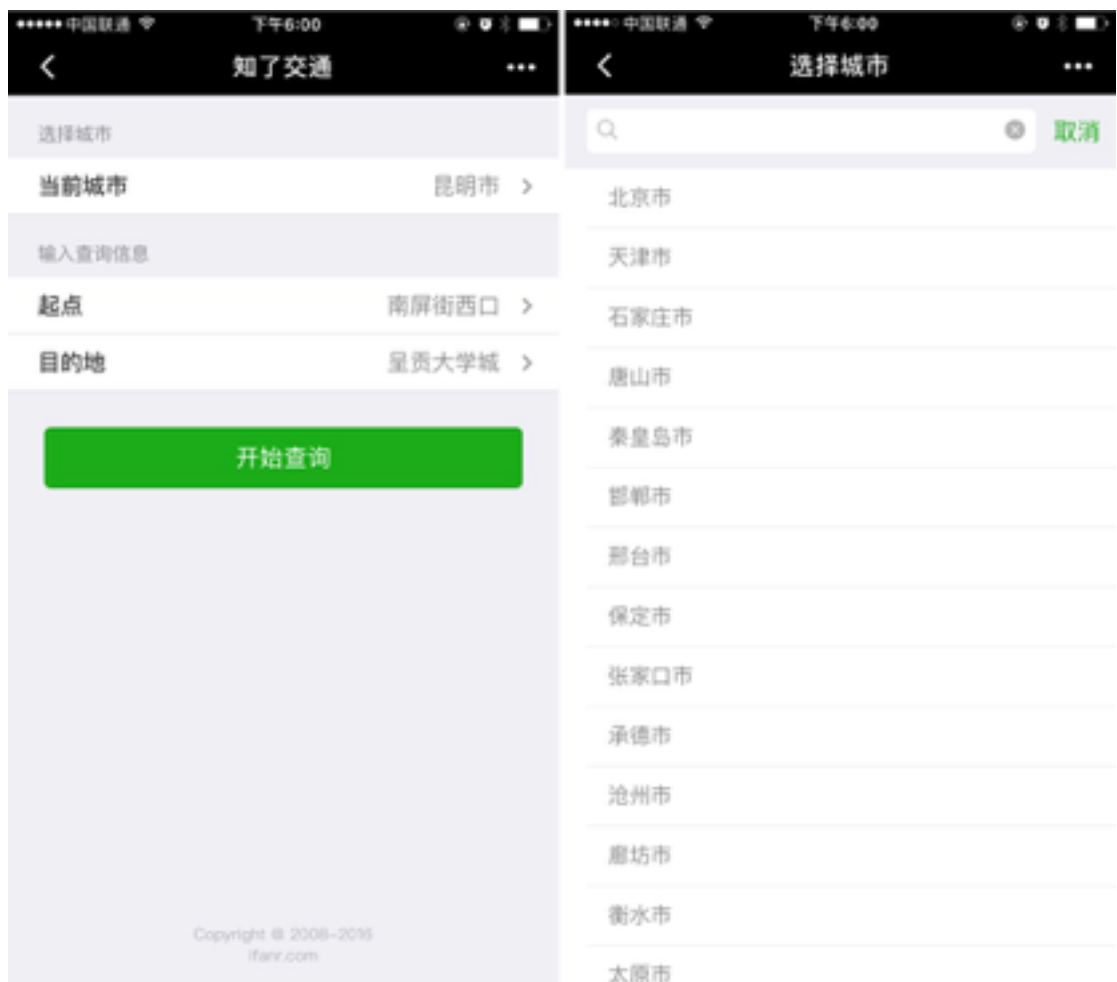


图 3-1-1 「知了交通」中选择城市和起始点的界面

点击「起点」、「目的地」，输入相应的位置，就可以开始查询从起点到目的地的推荐公交路径。

「知了交通」接入了百度地图的 API。因此，查询结果和百度地图是一致的，只是界面更为精简。

查询结果有「推荐路线」、「步行少」、「换乘少」、「地铁优先」四种，每一种方案都有具体的路线选择。

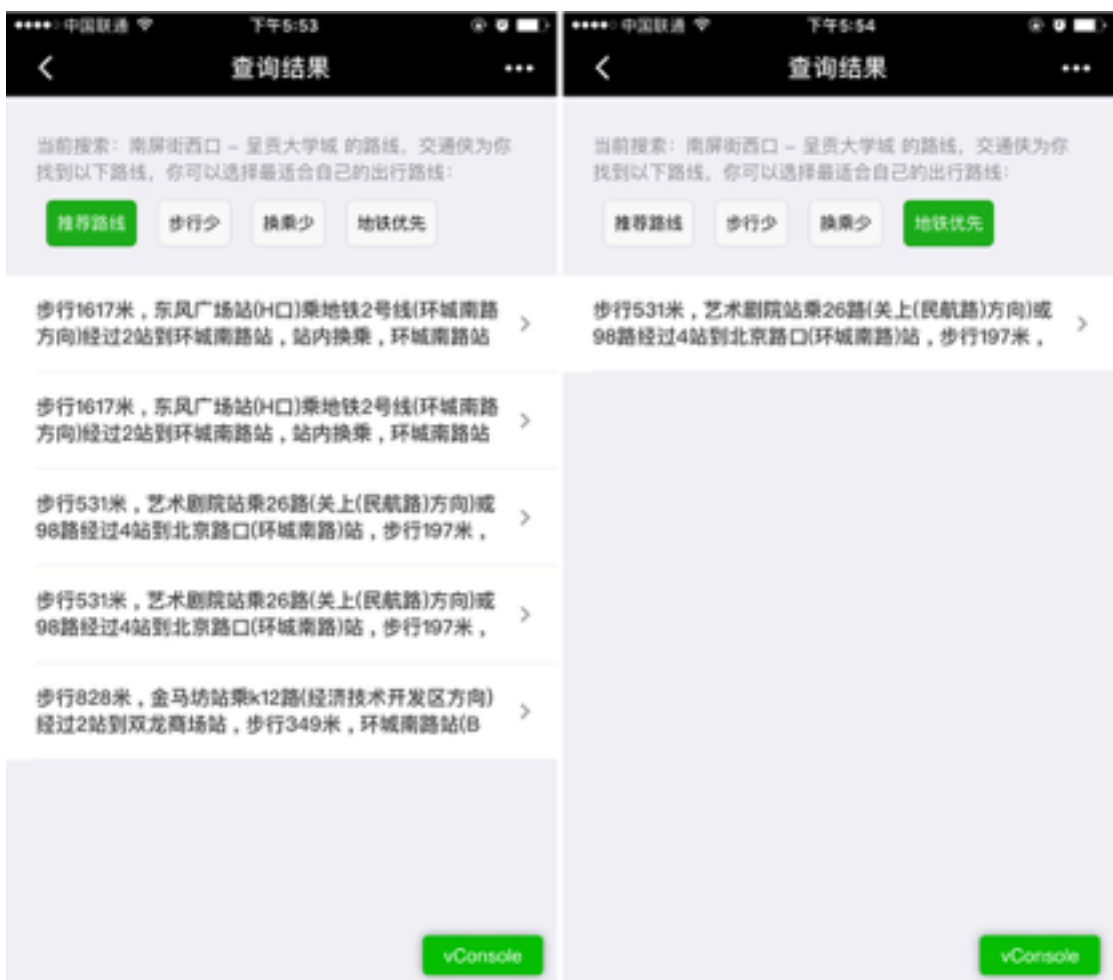


图 3-1-2 「知了交通」中，「推荐路线」和「地铁优先」路线比较

点击某一条具体路线，就会进入路线详情页，其中明确标识了起始的站点，以及步行距离。

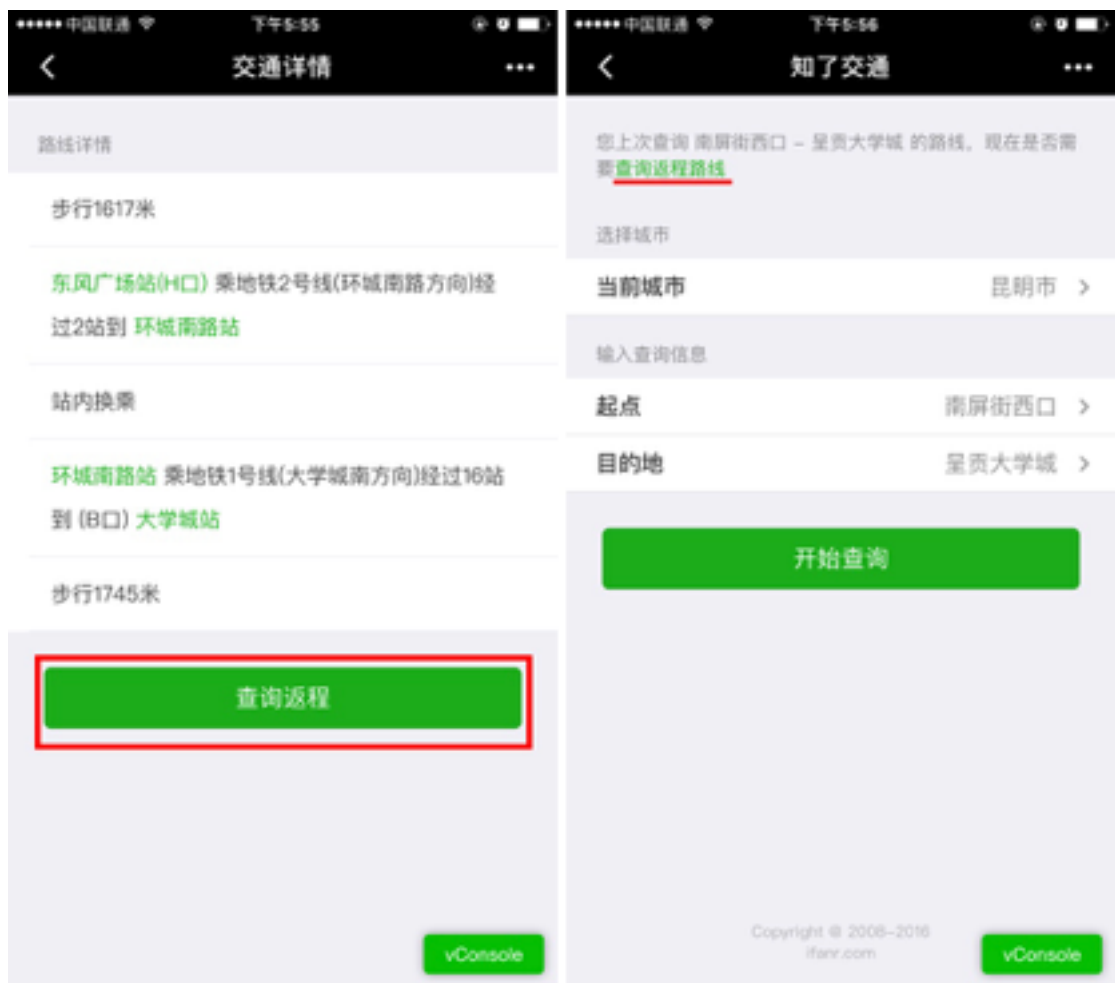


图 3-1-3 「知了交通」路线查询结果

「知了交通」还提供了非常方便的返程路线查询。在每一个路线详情页下方，都有「查询返程」的按钮。

除此之外，「知了交通」还会自动记录下你的上一次查询，当你回到首页，在顶部就能看到上次查询的路线，点击「查询返程路线」即可方便地查询返程。

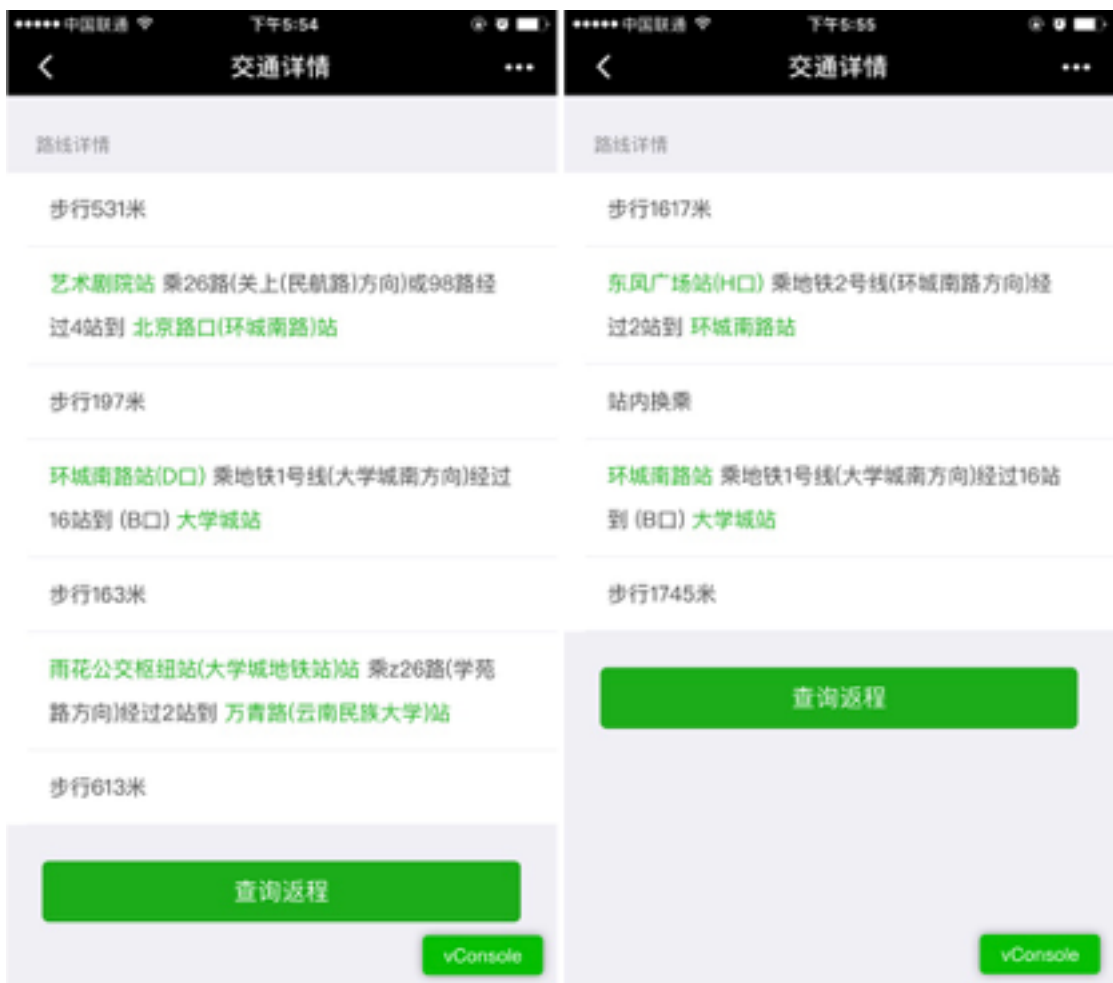


图 3-1-4 查询结果中的「查询返程」功能

值得注意的是，「知了交通」也可以自动获取你的位置。只要在进入首页时，在弹出的信息框中，点击允许「知了交通」获取你的地理位置，即可轻松设置所在城市和起终点位置。

二、为什么要开发「知了交通」？

「知了交通」的开发者是爱范儿的前端开发工程师严灏。他认为，这款小程序会让交通查询更加方便：

微信是大家每天都会开着的 app，有了「知了交通」，以后查路线就不用专门去打开一个新的 app，直接从微信进入就可以了。

他还表示，现在有很多交通/地图 app 在使用完退出后，会偷偷在后台运行，不仅偷跑流量，还会更耗电。

而「知了交通」这类轻量级的小程序，功能简单纯粹，也不涉及过多的体验。打开方便，用完即走，不耗电而且方便。

这些功能，基本上满足了大多数人日常交通查询的需求。

三、 小程序的开发经验

「知了交通」的开发大约用了 2 天时间，整个开发体验如何呢？

严灏表示，「知了交通」的功能十分简单，**所以整个开发过程很流畅，没有遇到什么问题**，比常规的前端开发简单很多。

谈到小程序开发和前端开发的差异时，严灏强调道，小程序和前端是不一样的东西，就像微信官方所说的一样，小程序不是浏览器，而是有自己的一套逻辑。所以，在开发过程中需要转变思路。

微信小程序目前还处于公测阶段，有很多 bug。出于安全性和某些其他考虑，小程序也只开放了一些组件，很多功能都开发不出来，也有很多东西需要进一步完善。

严灏认为微信小程序优点和缺点都是存在的：

如果是当前小程序框架范围内的东西，开发起来会比较简单轻松；如果需求超出目前的框架，那么，就需要自己造轮子，比较麻烦了。

剁手吗

知晓程序（微信号 [zxcx0101](#)）要分享的第二款小程序，是爱范儿前端女王大人开发的「剁手吗」，有了这款小程序，就能迅速知道看中的苹果产品，当下值不值得买。

一、「剁手吗」功能详解

相信除了资深果粉，很少有人会知晓苹果的每一个产品，更别说它们的发布时间了。

那么，当想要购买一款苹果产品时，你会怎么开始？去苹果官网？刷 **Apple Store**？或者，试试下面这个更简便的方式。

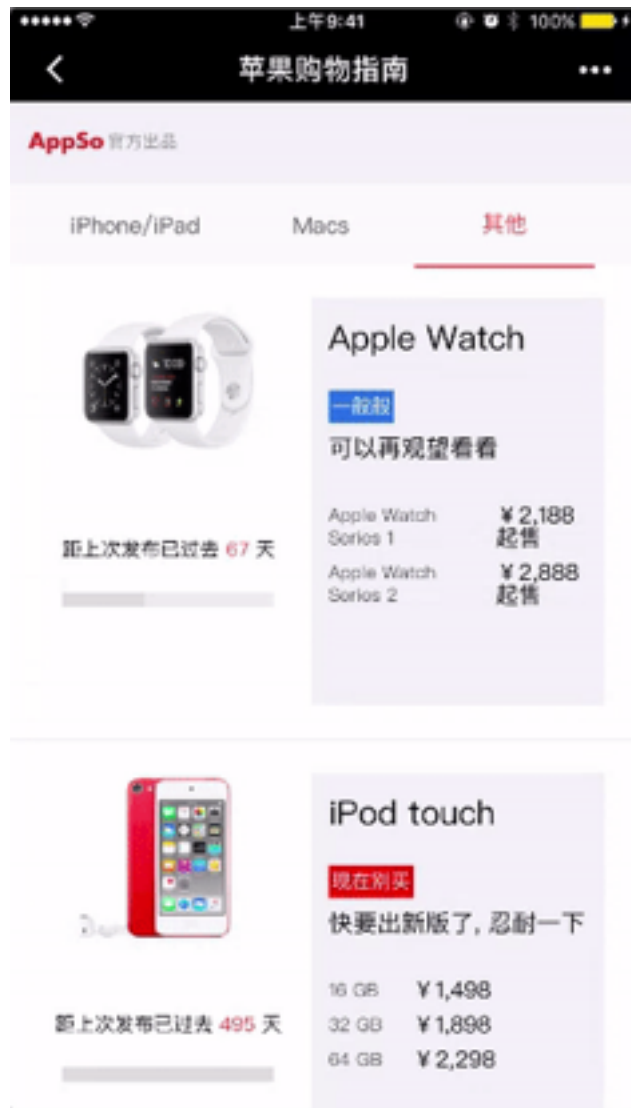


图 3-2-1 「剁手吗」

剁手吗是一款面向大众群体，提供苹果产品消费指南的小程序。里面搜集了苹果官网在售的所有产品（不包括配件）。

这款小程序的界面非常简明：进入首页，所有在售的 iPhone/iPad 尽收眼底。

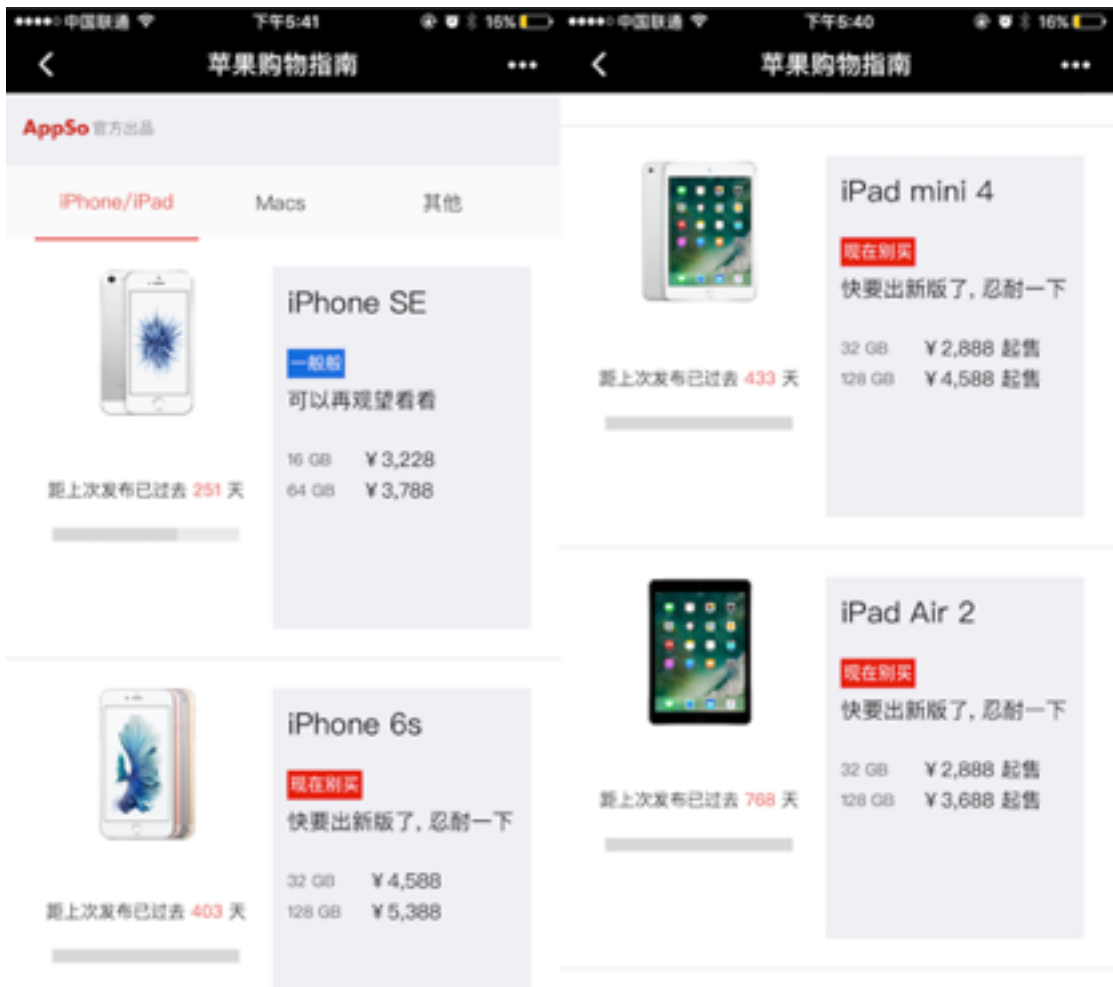


图 3-2-2 「剁手吗」iPhone/iPad 商品列表

点击顶部导航栏的「Mac」，就进入了 Mac 在售产品列表页面；
点击「其他」，就能看到 Apple Watch, iPod touch 等产品。

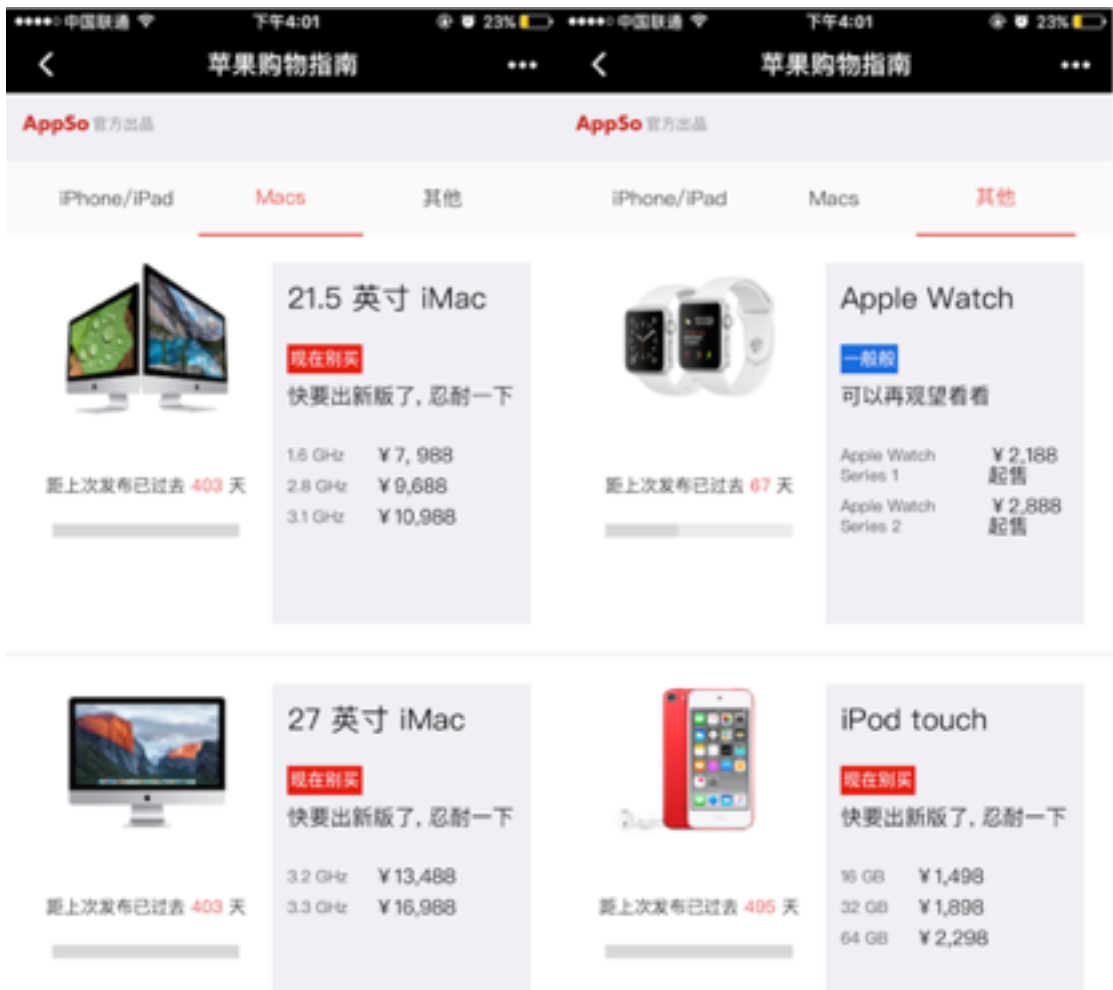


图 3-2-3 「剁手吗」Mac 及其他商品列表

这样，苹果官方渠道现在可以买到哪些产品，就一目了然了。

发现感兴趣的产品，点一下，就能进入产品信息页面。

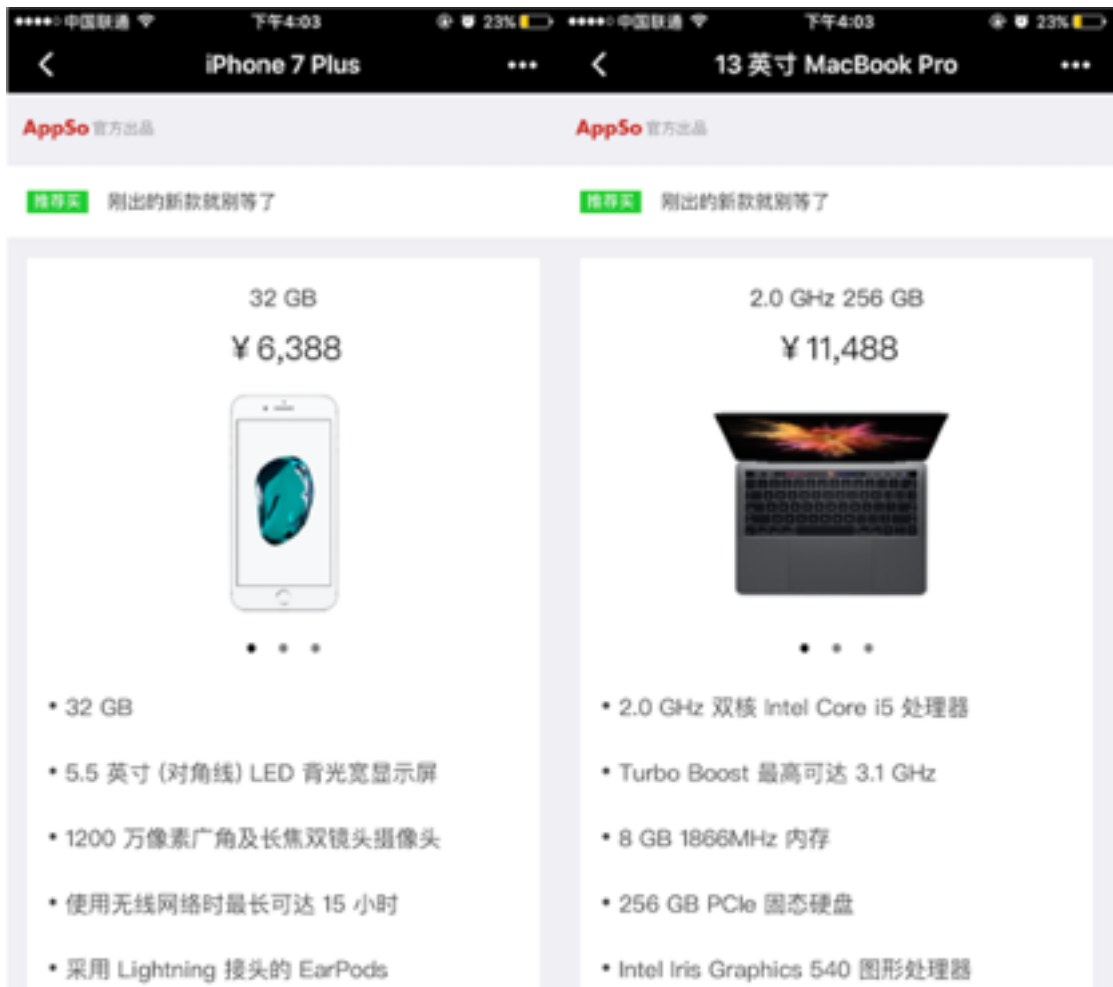


图 3-2-4 商品详情页

页面精选了产品的关键信息，就像在 **Apple Store** 看到的产品信息卡片一样。不用担心看到一大堆头昏眼花的详细参数。

以上这些信息，展示的主要是「有哪些」苹果产品，它们分别「是什么」，但究竟该买不买呢？

剁手吗有一个非常贴心的功能，就是每个产品有一个发布进度条和发布时长。

发布时长指的是距离产品发布日，已经过去了多少天；而发布进度条则综合了发布时长和产品的发布周期，你就可以推算出离下一代新产品发布日还有多久。

进度条越长，就代表离新产品发布的时间越近；进度条如果到顶了，就代表下一代产品已经发布了。

这样，就能非常直观地看出你感兴趣产品的新鲜程度。

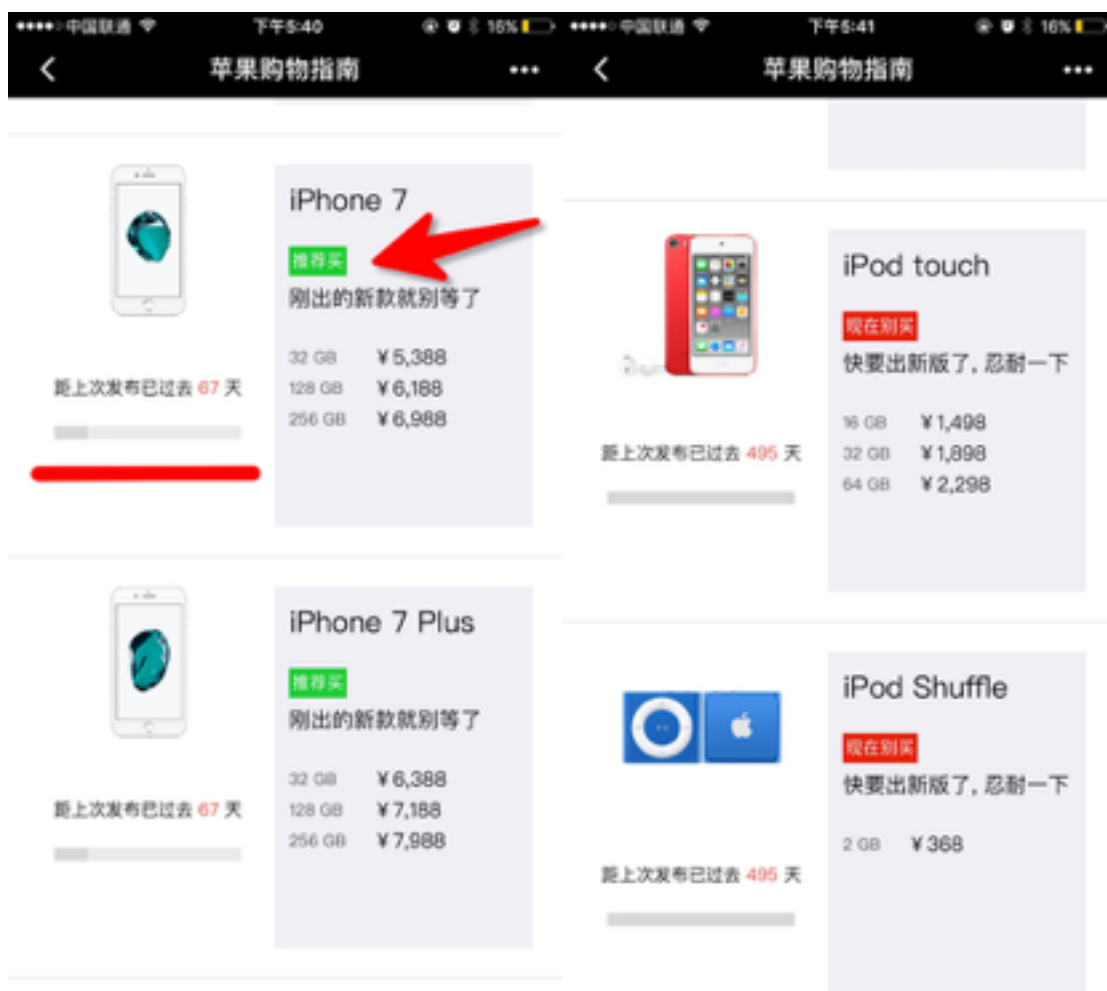


图 3-2-5 新鲜度显示和「推荐买」功能

知晓程序（微信号 zxcx0101）最喜欢的功能是，剁手吗给每个产品都提供了「购买建议」，以彩色标签的形式呈现：

推荐买：刚出不久的新款，赶紧下手。

一般般：这个时间点，可买可不买。

现在别买：新款快出了/新款已经出了，去买新款吧。

俗话说，电子产品买新不买旧，有了剁手吗，你就能快速知道产品的发布时间和周期，从而判断这款产品现在值不值得下手。

二、为什么要开发「剁手吗」？

正如上文所述，普通用户在选购苹果产品的时候，一时之间，可能会不知道该如何做选择。

去官网详细比对各项功能，又是一件比较耗费时间和精力的事。

要去比对什么呢？对于很多用户来说，他们其实更想要一个明确的指引：在这个时间，某款产品到底值不值得买？

而剁手吗这款小程序的设计，便能帮助这些用户迅速达成消费决策。

三、「剁手吗」开发经验分享

剁手吗的开发者，是爱范儿的前端女王大人祝莎莎（据说跪在她旁边写代码不会出 bug）。

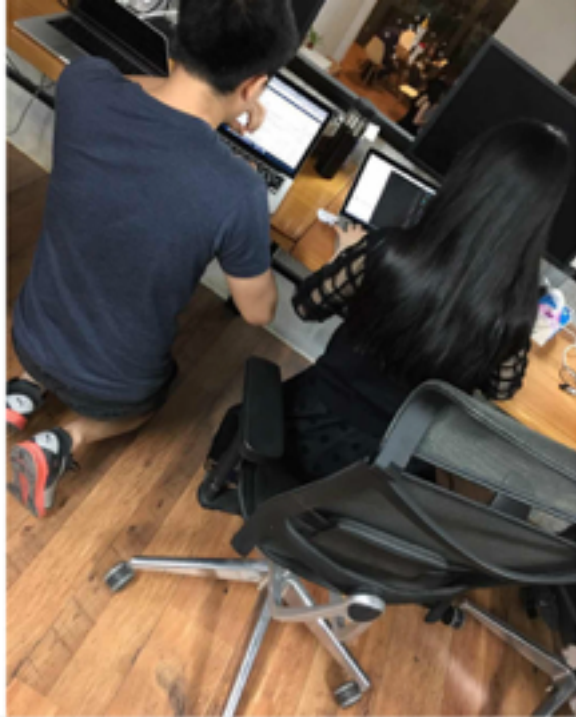


图 3-2-6 祝莎莎与同事的工作场面

由于调用的是静态数据库，所以，莎莎只花了一个下午的时间，就完成了这款小程序的开发工作。

说到小程序的开发体验，莎莎表示，整个框架写起来很顺手，体验也非常好，基本上没有遇到什么问题。

在所有小程序的开发过程中，她遇到的问题基本上是组件问题：

全局组件的封装都需要自己来做，比较麻烦，希望微信能尽快解决这个问题。

莎莎在采访中透露，自己未来也倾向于去做小程序的开发工作。

因为，微信小程序不需要考虑兼容性的问题，开发起来很畅爽。

此外，与常规前端开发工作不同的是，小程序更加简洁、朴实，且聚焦于功能，没有太多花里花哨的交互。这样，产品狗再也不能以 DOM 为理由，提一些任性的要求了。

知了地铁

知晓程序（微信号 **zxcx0101**）要给大家介绍的第三款小程序「知了地铁」，可以让你快速查询全国城市地铁线路。

一、「知了地铁」功能体验

「知了地铁」也是一款提供了全国地铁线路图参考的小程序。它的功能非常简洁：

一进入首页，你就能看到当前所在城市的地铁线路图。点击上方的城市栏，还能切换国内其他城市。



图 3-3-1 地铁图和选择城市界面

目前，「知了地铁」集合了全国 23 个城市的最新地铁线路图，不论你去到哪个陌生城市，都不用担心会地铁站里迷路。

在「知了地铁」中，你可以对地铁线路图进行缩放和移动。



图 3-3-2 在「知了地铁」中缩放地图

「知了地铁」提供了两种常见的缩放方式：

- 双指手势缩放
- 双击缩放

但囿于微信小程序在元素实现方面还有 bug，目前缩放功能的体验还不能做到特别完美。

对于生活在地铁网密集的大城市的人来说，「知了地铁」算是一款轻量且实用的小程序。

二、为什么要开发「知了地铁」？

爱范儿的产品总监李本卿表示，当张小龙宣布要推出应用号（小程序）时，就有打算做一款能让用户「用完即走」的产品。

而「知了地铁」正是一款这样简单实用的小程序，它能让人在乘坐地铁时，轻松了解一个城市的整体地铁线路图。

「知了地铁」的开发者的严灏也认为这是一个很实用的工具。以前他在深圳时，地铁卡的背面就有深圳地铁线路图，但广州的羊城通（地铁卡）上并没有。

所以，当去一个要转几次地铁线路的目的地时，查找路线就变得非常麻烦。严灏认为：

也不需要去打开地图 app，有一张线路图看看，问题就解决了。

三、「知了地铁」开发经验分享

「知了地铁」的开发过程并不顺利。

开发者严灏表示，整体开发工作其实一天就完成了，但为了实现更好的效果和体验，他又花费了一两周的时间，修复手势缩放的问题。

他表示，在微信小程序的 API 中有获取用户触摸屏幕事件的函数，通过这个函数可以获取到用户触摸屏幕坐标的对象。而通过持续性捕捉坐标变化，小程序就能实现手势缩放。

但可惜的是，小程序目前还无法实现完美的手势缩放。

例如，在使用「知了地铁」时，会出现缩放点不在两只手指中心的情况。在移动或缩放地图的时候，界面也会闪动，甚至会出现「缩放不跟手」的情况。

严灏尝试了各种方法解决这个问题。

包括调整抓取间隔为 16 毫秒等方法都试过了，最终的效果却不尽如人意。

后来严灏发现，这个问题是微信小程序的 bug。

在文档里写的可以做都是「理论上可以做」，跟「实际上能做」完全是两码事。

之后，严灏为「知了地铁」新增了双击放大的功能，使用体验比手势缩放稍微好一些。但若要实现完美的用户体验，还需要等微信团队来修复。

解忧室

知晓程序（微信号 **zxcx0101**）要分享的第四款小程序，能帮你解决手机使用中的各种疑难杂症，它的名字叫「解忧室」。

一、「解忧室」功能详解

熟悉 **AppSo**（微信号 **appsolution**）的读者，肯定对「**A 君解忧室**」这个栏目名字非常熟悉。

而「解忧室」小程序，就集合了「**A 君解忧室**」的所有内容。

进入「解忧室」，你能看到各种在手机使用过程中会遇到的问题。例如「就寝功能失灵怎么办」、「不搭梯用 **Chrome** 插件等问题」等。

点击「解忧室」底部的「分类」按钮，你还可以获得所有问题的分类：「通用技巧」、「**App** 推荐」、「**iOS** 技巧」、「**Mac** 技巧」和「**Android** 技巧」，定向索引你的问题。

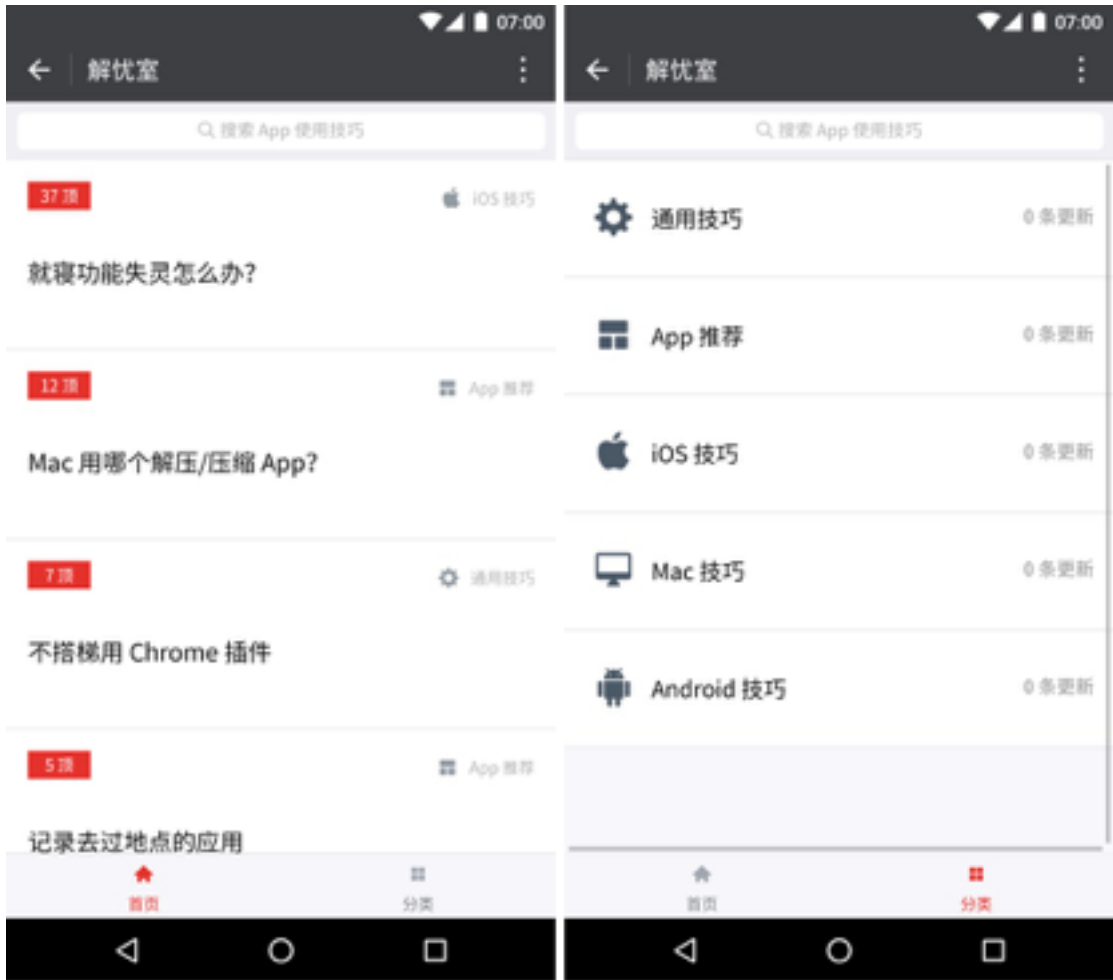


图 3-4-1 问题列表和分类

如果要寻找特定问题的解决方案，你也可以搜索关键词。「解忧室」小程序会帮你从已有问题中进行检索，快速找到解决方案。

点击具体的问题，就能进入到问题的解答页。如果这个回答帮助你解决了问题，你可以在解答页中点击「顶」按钮为它投票。



图 3-4-2 搜索和答案界面

投票的意义其实并不止于一个数字。越多人点「顶」，说明这个问题有越多的人遇到过，并通过这个解答解决了问题。

为答案投票的同时，这条回答也有机会被更多人看见，帮助其他用户快速解决他们的问题。

二、「解忧室」的开发背后

「解忧室」是爱范儿旗下 AppSo（微信号 appsolution）的小程序产品。

AppSo 负责人王崇旭表示, AppSo 有一个专门解答读者关于手机、电脑使用疑问的栏目, 叫「A 君解忧室」, 「解忧室」小程序的内容便来源于此。

我们都会收到很多读者在数字生活(使用数码产品)中遇到的问题, 每周我们会汇总起来做一次集中解答。现在「A 君解忧室」已经做了 30 多期, 每期都有 20 条左右的内容。

那么, 为什么要做一个「解忧室」小程序呢? 王崇旭说, 「其实都写在小程序的名字上了。」

出发点其实跟我们做「A 君解忧室」栏目是一致的。每个人都或多或少经历过用数码产品遇到问题、但找不到答案的窘境。如果你身边恰好有一个达人帮你解决, 那种感觉是很好的。

我们希望 AppSo 成为每个人都能依靠的「达人朋友」, 在遇到科技问题的时候可以在「解忧室」找到自己需要的答案。正因如此, 如果「解忧室」小程序真的能帮大家「解忧」, 我们就很满意了。

熟悉 AppSo 的人都知道, AppSo 目前已经有了 iOS 与 Android 客户端。

但是, 既然有了客户端, 为什么不直接将「解忧室」功能放到客户端, 而要做成一个小程序呢?

王崇旭认为, 小程序可以为客户端试错。

做 app 要考虑的事情太多了。现在 AppSo 有 iOS 和 Android 两个客户端, 如何将「解忧室」融入现有的产品, 并提供合理、良好的用户体验, 是需要谨慎考量过后才能决定的。

再者，我们也并不知道用户是否真的需要它变成客户端的一个功能，贸然开发也要承担不小的风险。

小程序开发成本低，是很好的试错机会，如果反响好，我们再将它移植到 app 上也不迟。

如果你玩手机时也遇到了一些无法解决的问题，那么，等到小程序正式发布以后，就来找「解忧室」吧。

第四节 如何转行小程序开发？

现在人人见面必谈小程序。这种火热让很多人慌了神：我们是不是一定要学小程序开发呢？小程序开发应该从哪里入手？

知晓程序(微信号 [zxcx0101](#))为你总结了小程序开发所需的技能，以及不同种类的程序员该如何转向小程序开发。希望能帮大家顺利开启小程序开发之路。

一、 必备技能清单

如果你想开发一个应用，那么了解和学习应用所用的编程语言是非常必要的。

对于小程序开发者来说，需要学习的语言主要有：

- 微信标记语言（WeiXin Marked Language, WXML）
- 微信样式表（WeiXin Style Sheet, WXSS）
- JavaScript（JS）

虽然微信为小程序的界面构建「创造」了两种语言，但这两种语言，无论是从风格还是语法上，都与 **HTML** 和 **CSS** 非常接近，甚至是直接照搬。

也就是说，在开发微信小程序之前，你需要掌握的语言只有 **HTML**、**CSS** 和 **JavaScript** 三种，而这三种恰恰就是网页前端天天都要面对的。

学习、开发过网页前端的人都知道，「前端三件套」的学习成本其实是很低的，有些人甚至能在一周内熟练掌握这三种语言。

二、如果你是 iOS / Android 开发人员

如果你以前有 **app** 开发经验，那转向小程序开发并不难。你只需要简单学习三种语言的写法，就能轻松上手开发小程序。

无论是从 **iOS** 还是 **Android** 开始转，开发小程序都需要学习 **JavaScript** 语言。小程序的逻辑功能都是使用 **JavaScript** 来实现的，如果没有 **JavaScript**，小程序就变成了一具没有思维的空壳。

对于 **iOS** 开发来说，可能还要学习使用 **HTML (WXML)** 构建界面的方式。

许多人在开发 **iOS** 应用时，会使用 **iOS** 的 **Storyboard** 文件构建界面和交互逻辑，**Storyboard** 可以让开发者通过拖动元素的方式快速构建界面。

但在小程序中，所有视觉元素和交互都需要通过 **XML (WXML)** 代码实现，无法直接拖动元素来构建界面。这与 **iOS** 的 **Storyboard** 可视化的界面构建方式有很大差别。

对于原本就需要 **XML** 代码构建界面和交互的 **Android** 开发者来说，他们只需要了解小程序 **WXML** 可以使用的元素，就可以开工了。

三、 如果你是网页前端开发人员

恭喜你，既然你已经能熟练使用 HTML、CSS 和 JS 语言，接下来的事情就简单得多了。

但这并不代表前端网页开发与小程序开发没有差别，你在上手前还是有一些需要注意的地方。

首先，小程序的界面并非是一个「网页」，而是一个「应用界面」，以往网页前端常见的 `<div>`、`<a>` 等标签都不复存在。网页前端开发人员需要适应 WXML 的逻辑和写法。

同样，虽然 WXSS 文件基本「复刻」了 CSS，但小程序使用了 rpx（responsive pixel）作为尺寸单位（保证不同分辨率设备的元素正常显示）。开发人员需要了解 rpx 与各种长度单位的换算比例。

微信关注 [zxcx0101](#)，回复「rpx」，获取 rpx 的详细换算方式。

四、 如果你是后端开发人员

即使是使用 Node.js 语言工作的后端开发人员，如果希望转向小程序开发，依然需要先从前端三件套入手学习。

不用担心，前端三件套是非常简单的语言，许多人只需要经过一周的学习就能熟悉。如果之前有相关经验，上手难度会更加低。

但是，许多业务流程完整的小程序是离不开后端的数据处理的。后端开发者并不需要太担心，小程序的火热会造成后端领域的侵蚀。相反，小程序的火热会令后端人员的需求迅速增长。

以上就是知晓程序（微信号 [zxcx0101](#)）为不同领域的开发者总结的小程序开发技能清单。希望能够帮助大家从不同领域快速上手小程序开发。

第五节 如何拥有自己的小程序？

想拥有自己的小程序，你需要做些什么？

微信官方提供的小程序接入指南列出了以下事项：

1. 小程序注册；
2. 小程序信息完善及开发前准备；
3. 开发工具的使用；
4. 审核和发布小程序；
5. 小程序申请微信认证；
6. 小程序申请微信支付；
7. 小程序绑定微信开放平台账号。

在以上事项的操作过程中，有什么值得注意的地方呢？在本节内容中，知晓程序（微信号 **zxcx0101**）提供了一份更为精简的教程。

如何注册微信小程序

11 月 3 日，微信公众平台发布了微信小程序开放公测的消息。



图 5-1-1 微信官方的小程序介绍页面

在微信官方给出的图片中，能看到小程序的简单介绍和注册指引。



图 5-1-2 微信官方提供的接入流程和开发支持

下面，知晓程序（微信号 [zxcx0101](#)）就教大家，如何一步步注册微信小程序。

首先，打开微信公众平台 <https://mp.weixin.qq.com/>，在右上角点击「立即注册」。



图 5-1-3 注册小程序的入口

接着，你会进入注册账号类型的选择页面。

可以看到，除了常规的订阅号、服务号和企业号之外，还出现了「小程序」的选项。



图 5-1-4 注册公众帐户的页面多了一个「小程序」

注册小程序只需完成三个步骤：

1. 填写账号信息

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](#)

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》

图 5-1-5 填写帐户邮箱、密码

2. 完成激活邮箱



图 5-1-6 验证邮件

3. 进行信息登记

用户信息登记

微信公众平台致力于打造真实、合法、有效的互联网平台。为了更好的保障你和广大微信用户的合法权益，请你认真填写以下登记信息。为表述方便，本服务中，“用户”也称为“开发者”或“你”。

用户信息登记审核通过后：

1. 你可以依法享有本微信公众平台所产生的权利和收益；
2. 你将对本微信公众平台的所有行为承担全部责任；
3. 你的注册信息将在法律允许的范围内向微信用户展示；
4. 人民法院、检察院、公安机关等有权机关可向腾讯依法调取你的注册信息等。

请确认你的微信公众平台主体类型属于政府、媒体、企业、其他组织，并请按照对应的类别进行信息登记。点击查看微信公众平台信息登记指引。

主体类型 如何选择主体类型?

企业	政府	媒体	其他组织
----	----	----	------

企业包括：企业、分支机构、个体工商户、企业相关品牌。

主体信息登记

企业类型 企业 个体工商户

企业名称

需与当地政府颁发的商业许可证书或企业注册证上的企业名称完全一致。信息审核成功后，企业名称不可修改。

营业执照注册号

请输入15位营业执照注册号或18位的统一社会信用代码

图 5-1-7 基础信息登记界面

OK，完成注册！整个过程非常简单顺畅，用时不到 3 分钟。

需要注意的是，目前微信小程序仅支持企业、媒体、政府和其他组织注册，暂不支持个人注册。

想第一时间体验小程序？就赶紧打开电脑注册吧！

如何搞定微信认证？

微信小程序开放公测之后，很多组织都注册了自己的小程序。注册时，也有不少人用微信认证的方式来确认主体身份。

但是，很多人在第一次进行微信认证时，会踩中很多坑：资料认证不全、与其他同事联系沟通不畅等，都会导致微信认证过程出现多次返工，甚至认证失败的情况。

现在，知晓程序（微信号 **zxcx0101**）就为大家带来一份微信认证的全过程指引。只要一步步按着指引来做，就能轻松完成小程序认证。

一、小程序必须认证吗？

有很多人以为，主体为组织而非个人的小程序需要经过微信认证，才能正常审核和发布。

然而，《微信小程序接入指南》中是这样写的：

「政府、媒体、其他组织类型帐号，必须通过微信认证验证主体身份。企业类型帐号，可以根据需要确定是否申请微信认证。已认证帐号可使用微信支付、微信卡券等高级权限。」

换言之，如果小程序的主体是企业，且小程序不需要使用到支付和卡券相关的功能，则可以跳过申请微信认证。

只需要在注册过程中，选择「向腾讯公司小额打款验证」这种方式，就可以跳过微信认证步骤进行主体验证。

主体信息登记

企业类型 企业 个体工商户
企业包括：企业、分支机构、企业相关品牌等

企业名称
需与当地政府颁发的商业经营许可证或企业注册证上的企业名称完全一致。信息审核审核成功后，企业名称不可修改。

营业执照注册号
请输入15位营业执照注册号或18位的统一社会信用代码

注册方式 向腾讯公司小额打款验证 微信认证

① 填写企业对公账户
为验证真实性，此对公账户需给腾讯打款验证。注册最后一步可查看打款信息。请尽快联系贵公司/单位财务进行打款。

微信认证
微信注册并认证，无需小额打款验证，需支付300元审核费用。提交认证后会在1-5个工作日完成审核。在认证完成前小程序部分能力暂不支持。 [查看详情](#)

图 5-2-1 申请打款验证的入口

二、 准备资料

在认证开始之前，你需要准备一堆资料。

根据组织性质的不同，所需要准备的资料也是有一定差异的。

根据微信官方的说明，可以通过微信认证的组织有以下几类：

- 企业法人
- 媒体
- 政府及事业单位
- 基金会、外国政府机构驻华办事处
- 社会团体
- 民办非企业
- 其他组织

不同性质的组织，认证所需要的资料分别为：

主体类型		所需资料
企业法人		组织机构代码证 企业工商营业执照
媒体	所有	组织机构代码证 企业工商营业执照副本 或 事业单位法人证书
	广播电台	广播电视播出机构许可证 或 广播电视播出许可证
	报纸	中华人民共和国新闻出版许可证
	期刊杂志	中华人民共和国期刊出版许可证
	网络媒体	互联网信息服务许可证 或 信息网络传播视听节目许可证
政府及事业单位		组织机构代码证
免费认证机构 ^(*)	所有	组织机构代码证
	基金会	基金会法人登记证书
	外地登记机构	所在地政府主管部门的社文
	外国驻华机构	国家有关主管部门的社文或证明
	地方委员会 ^(*)	主管部门的社文或证明
独立地位的附属机构	主管部门的基本存款账户开户许可证和批文	
社会团体	所有	组织机构代码证 社会团体登记证书
	非营利组织	原业务主管部门的社文或证明
民办非企业	所有	组织机构代码证 民办非企业登记证书
	非事业性质的培训机构	民办非企业办学许可证
	医疗机构（包括民营）	民办非企业医疗机构执业许可证
其他盈利组织		组织机构代码证 相关登记证书、社文或证明等

注：
*1：微信官方的分类称为「其他组织-免费」。这类组织进行微信认证时不需要缴纳300元的认证服务费。
*2：包括村民委员会、村民委员会、社区委员会等。

图 5-2-2 认证所需资料一览

三、 填写资料

小程序的主体信息登记完成，管理员就可以在公众平台中对小程序进行认证。

使用小程序帐户登录公众平台后台，在左侧点击「设置」，在基本信息中找到「微信认证」一项，点击左侧「详情」，就可以找到认证入口。

认证前，管理员需要同意一份服务协议。阅读协议（真的会有人认真阅读吗...）并勾选同意，就可以开始进行认证。

首先，需要准备认证公函。

在资料填写部分，可以找到公函的下载地址。你需要做这些：

下载打印公函模板；

按要求填写公函内容、盖章；

对公函进行拍照或扫描（需彩色件）；

在页面中上传公函文件。

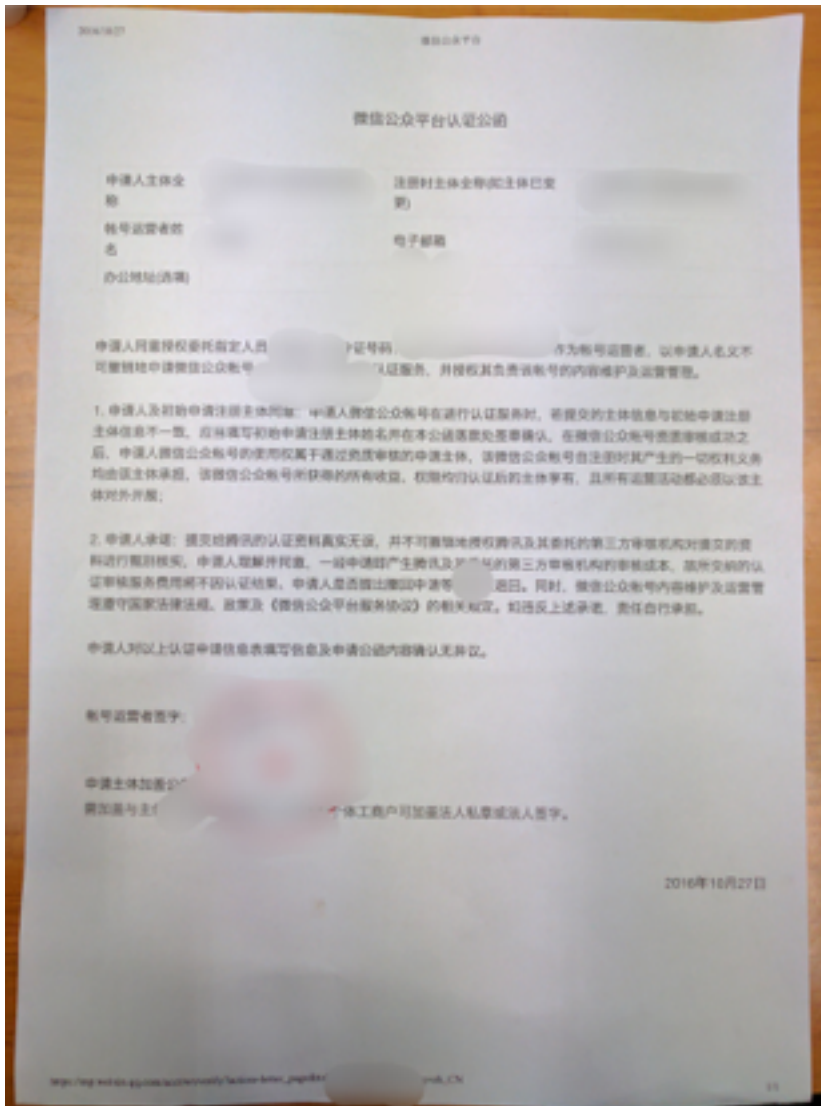


图 5-2-3 认证公函示例

之后，就可以继续填写其他资料了。

在资料填写页面，还需要填写认证所需的**证件号码**；上传**证件扫描件、管理员身份证的正反面文件**。

如果有开具发票的需求，可在填写完资料，并验证了管理员身份后申请开具发票。

填写发票

1. 发票资料提交后不能修改。请填写正确的发票类型和寄送地址，若填写错误造成发票开具错误、寄送错误或选择不开发票，后续将无法重新开具并寄送发票。
2. 普通发票为企业发票，不显示发票抬头；增值税专用发票的抬头为认证申请机构的全称。
3. 因三证合一问题导致税号变更的，请确认需要报税的税号再填写，否则发票开具信息错误将无法重新开具。
4. 审核结束后（包括认证成功和失败），腾讯会在45天内开具并寄出普通发票，80天内开具并寄出增值税专用发票。

发票类型	增值税专... 企业必须有一般纳税人资质，且提交《税务登记证》、《银行开户证明》给腾讯客服核对无误后方可开具。
发票抬头	叶生牙科 该名称为申请注册时填写的企业/媒体单位/组织名称。
纳税识别号	<input type="text"/> 请填写15位、17、18或20位税务登记号，如超过位数则无法开具专用发票 因三证合一问题导致税号变更的，请确认需要报税的税号再填写，否则发票开具信息错误将无法重新开具。
企业注册地址	<input type="text"/> 企业注册地址 与税务登记证一致
企业电话	<input type="text"/> 企业电话
企业的开户银行	<input type="text"/> 测试 与银行开户证明一致

图 5-2-4 发票申请页面

如果需要开具增值税专用发票，还需要填写额外的**税务登记证**和**银行开户证明**。

四、 费用支付

进行一次微信认证服务，需要向第三方公司缴纳 **300** 元的资质验证费用（基金会、外国政府机构驻华办事处等除外）。**值得注意的是，认证服务费用的支付只能使用微信支付。**

微信官方的说明是：这个费用是提交给资质审核机构的一种真实性验证的服务费用。由于审核行为定义为服务，所以无论审核成功与否都需要缴纳费用，并且缴纳费用不代表审核一定可以通过。

支付完毕后，审核服务即宣告开始。

五、 审核期间

审核服务订单提交以后，在公众平台后台的通知中可以看到参与审核服务的企业的联系方式。管理员需要在审核期间保持通讯畅通，以保证能收到审核人员的电话进行资料核对。

根据知晓程序（微信号 **zxcx0101**）的经验，审核人员会询问企业法人姓名、公司注册地址（区域），以及对公账户所收到的具体打款金额这些信息。

如果在电话核对过程中，出现了资料错误或遗漏等问题，则需要补充或重填。微信官方提供了 30 天内 3 次补充和修改机会，管理员可以在期间对审核资料进行调整。

我们在认证小程序的时候发现，从提交到电话审核，最快半天就可以认证成功。

六、 认证年审

小程序的微信认证有效期为一年。在认证有效期结束之前，如果小程序需要延续其认证身份，则需要重新进行认证。

重新认证的流程与上述流程一致，值得注意的是，**重新认证也需要缴纳资料审核服务费用。**

以上就是微信认证的全过程解析，希望这篇文章能帮你迅速流畅地完成微信认证。

特别提醒：建议在阅读完本文后，根据所属的组织类型，先列出认证所需要的物料清单，再进行微信认证，以减少不必要的时间浪费。

如何完善小程序信息？

知晓程序（微信号 [zxcx0101](#)）已经分享了微信小程序的快速注册和认证教程。现在，我们就来看看，在小程序信息完善过程中，有哪些注意事项。

一、 你的小程序认证了吗？

很多人的小程序会选择微信认证来验证身份。

然而，别以为小程序通过认证后，微信会主动提醒你。

真实情况是，接到微信认证机构打来的电话后，就算小程序成功通过认证，你留的手机和邮箱也不会收到任何信息。

想知道自己的小程序是否通过认证，最好的办法是时不时登录微信公众平台看看。



图 5-3-1 认证服务开始通知

据微信认证机构的短信所示，他们「预计会在 2 个工作日内」受理。实际上，我们早上注册小程序，接完认证电话，下午悄无声息就通过了认证.....

二、填写小程序资料

小程序认证后，你就可以开始填写小程序信息了。

1. 小程序名称

小程序名称支持中文、英文、数字和加号，长度为 3 - 20 个字符。



图 5-3-2 小程序起名要求

值得注意的是，加号「+」只能出现在名称的最末，如「知晓程序+」；而「知晓+程序」、「+知晓程序」这类命名方式都是错误的。

如果你曾遭遇过订阅号、服务号「名称已被注册」的情况，那么，给微信小程序「起个名字」恐怕会令你更加抓狂：

- 不得与微信公众平台现有的订阅号、服务号重名。
- 若侵犯了其他企业商标和组织名称等，可能会被投诉，收回名称。
- 存在大量账号同名的名称也不给用，如「健康生活」。

要是不想改名，就善用这个加号「+」吧。友情提示，用「+」山寨别人的名字也是不行的哦。

目前而言，小程序一旦确认了名称，就不能再修改了。至于什么时候能修改，就要看微信怎么定了。所以，起名还请慎重。

2. 小程序头像

头像设置这个环节，小程序跟订阅号、服务号是一样的。



图 5-3-3 小程序头像设置界面

- 图片格式: bmp, jpeg, jpg, gif
- 图片大小: 不可大于 2M

此外，小程序头像不得出现政治敏感和色情的内容。每个月可修改 5 次。

3. 小程序介绍

字数为 4 - 120 个字，不包含国家相关法律法规禁止内容就行了。



图 5-3-4 小程序介绍设置界面

跟头像一样，小程序介绍每个月也有 5 次修改机会。

4. 服务范围

微信小程序共包含了 17 个可选服务范围，每个服务范围下面又有细分领域。每个小程序最多可以选择 5 个服务范围。



图 5-3-5 服务范围选择界面

小程序支持哪些服务范围呢？知晓程序（微信号 zxcx0101）特地整理了一张表格供大家查看：

服务范围	细分领域
金融	基金、信托、理财、保险、银行、股票信息服务平台、证券公司
商业服务	律师、律所、招聘、农林牧渔
公益	公益
快递、物流与仓储	快递物流、仓储
教育	成人教育、大学教育、婴幼儿教育、在线教育、培训机构、出国、中小学教育
富媒体	阅读、视频、语音、音乐、图片
工具	记账、投票、日历、天气、备忘录、办公、字典、图片
出行与交通	市内公交、长途客运、停车、代驾、租车、打车、飞机、火车、地图、轮船
房地产	商业地产、物业管理、开发商、房产交易、装修建材、地产观察与评论
生活服务	票务、生活缴费、家政、外卖、丽人、印刷、婚庆摄影、加油、养车、宠物、废品回收
IT 科技	《智能》硬件与设备、通信与运营商、软件与服务、手机营业厅
美食	点评与推荐、菜谱、餐饮与服务
旅游	旅游线路、酒店、公寓、门票、出境 WiFi
休闲娱乐	私人影院、网吧、采摘、农家乐、diy 手工坊、密室逃脱、棋牌室、桌球馆、真人 cs、桌面游戏、养生、茶馆、足疗按摩、洗衣桑拿、ktv
电商	女装/男装/内衣、鞋靴/箱包/配件、童车玩具/孕产/辅食、家电/数码/手机、美妆/洗护/保健品、珠宝/眼镜/手表、运动/户外/乐器、游戏/动漫/影视、美食/生鲜/零食、鲜花/宠物/农资、家居/家纺/家纺、汽车/二手车/商品、办公/DIY/五金电器、百货/餐厨/家居保健
医疗	健康咨询、就医服务、医疗机构、药店
政务民生	住房、人力资源、文化场馆、社会保障、图书馆、交通违法、博物馆、出入境

图 5-3-6 服务范围一览表

值得注意的是，若是选择了律师、医疗等服务范围领域，还需单独上传相关资质文件。

由表格不难看出，小程序的服务范围几乎涵盖了生活的方方面面。虽然，绝大多数人对现在对小程序还没有什么概念，但相信很快小程序就会成为我们日常生活的一部分。

开发者工具怎么用？

工欲善其事，必先利其器。

对于小程序开发者来说，微信提供的开发者工具是不可绕过的一步。

现在，知晓程序（微信号 [zxcx0101](#)）就来教你如何一步步设置好开发者工具。

一、 如何获取开发工具

开发小程序，需要用到「**微信 Web 开发者工具**」。

通过以下步骤，可以找到开发工具的下载地址：

1. 打开微信公众平台（[mp.weixin.qq.com](#)）；
2. 鼠标移至「小程序」，点击「开发」按钮；
3. 点击顶部导航栏的「工具」按钮，在新页面的左侧栏中找到「下载」。

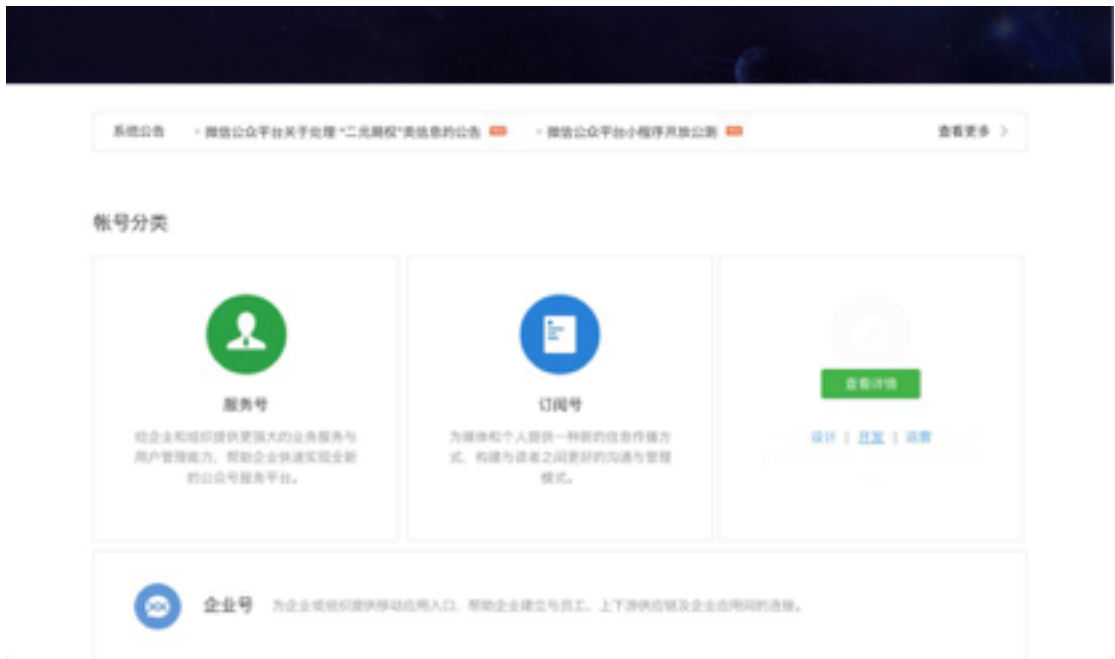


图 5-4-1 在微信公众平台主页找到「开发」链接

开发工具提供 Windows 32 位、Windows 64 位和 Mac 三个版本。可根据实际情况，选择适合自己电脑的开发工具。

以 Mac 电脑为例，只要将下载的 DMG 文件打开，并根据提示，将 app 文件移动至「应用程序」文件夹，安装就完成了。

二、 登录并新建项目

安装后，就可以直接启动开发者工具。

如果是首次启动开发者工具，需要使用开发者的微信号扫码登录。



图 5-4-2 登录至开发者工具

登录后，微信开发者工具会询问调试类型，请选择「本地小程序项目」。

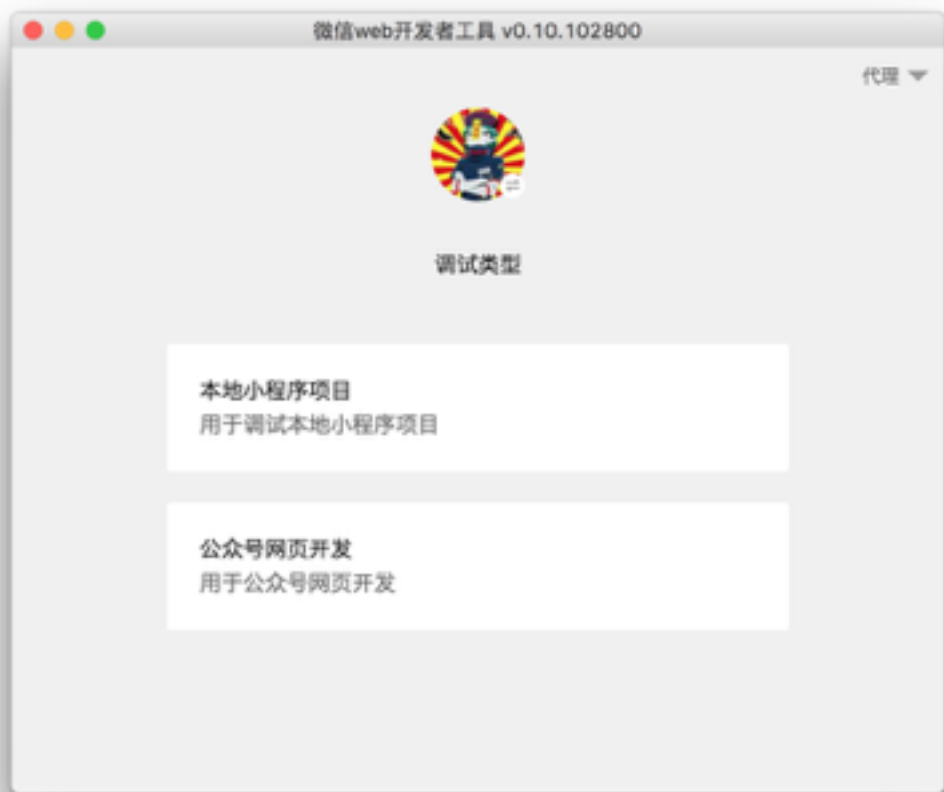


图 5-4-3 调试类型选择

之后，点击「添加项目」按钮，就可以在电脑上新建小程序项目了。

填写 **AppID**（如果没有，则点击「无 **AppID**」）、项目名称。接着，在项目目录中点击「选择」，新建一个文件夹作为新的项目目录。



图 5-4-4 添加项目

选择项目目录后，开发者工具会询问是否「在当前目录中创建 quick start 项目」。如果勾选，开发者工具会在相应文件夹下创建小程序的必要文件。

添加项目之后，以后打开开发者工具时，就可以在项目列表中找到新建的项目。



图 5-4-5 项目管理界面

点击项目就可以继续进行开发。

三、认识开发工具

微信开发者工具共分成「编辑」、「调试」和「项目」三个部分。

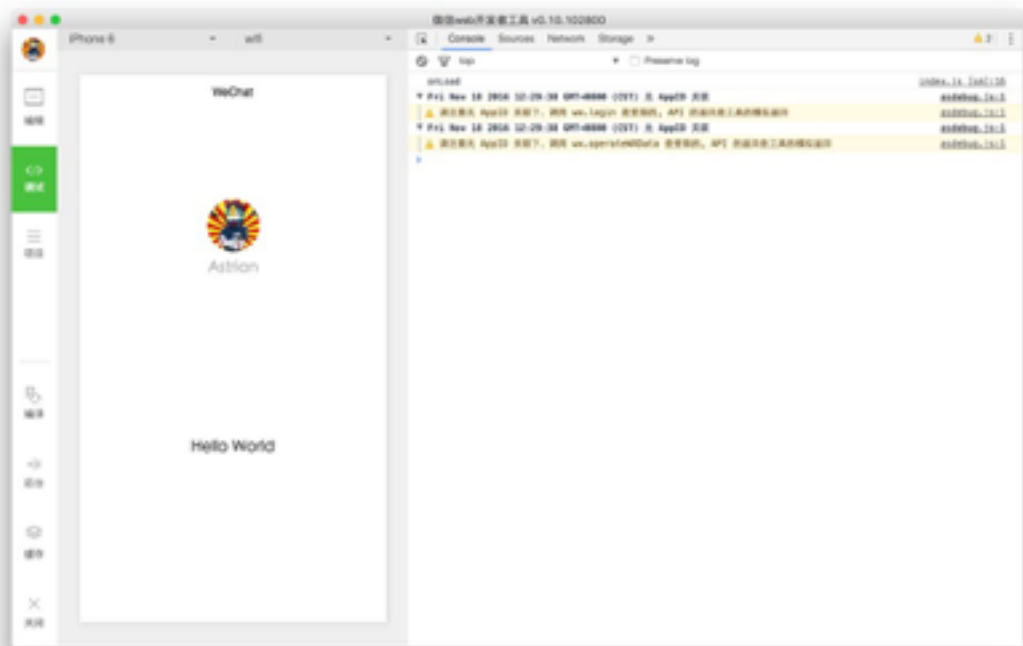


图 5-4-6 开发者工具界面

- 一、 **编辑**：编辑和修改小程序的代码。
- 二、 **调试**：显示小程序代码错误和警告，便于调试代码错误。
- 三、 **项目**：用于查看小程序的属性、修改小程序配置，以及提交小程序代码至微信服务器。

开发工具还提供简单的**模拟器**（页面左侧），开发者可以使用模拟器在电脑上模拟用户点击及使用，提供后台情况模拟、多媒体播放器调试、缓存数据处理等功能。

四、 真机调试和提交代码

在「项目」页面中，开发者可以查看小程序的基本属性，也可以进行真机预览调试，或向微信服务器提交一个新版本的小程序代码。



图 5-4-7 「项目」界面

想到真机上体验代码的运行效果？点击「项目」中的「预览」按钮，用当前开发者的微信号扫码即可。

如果需要在真机运行的时候进行调试，可以点击小程序右上角的「更多」按钮，选择打开调试。接着，重新扫码启动小程序，右下角就会出现一个「vConsole」按钮，点击即可打开调试器。

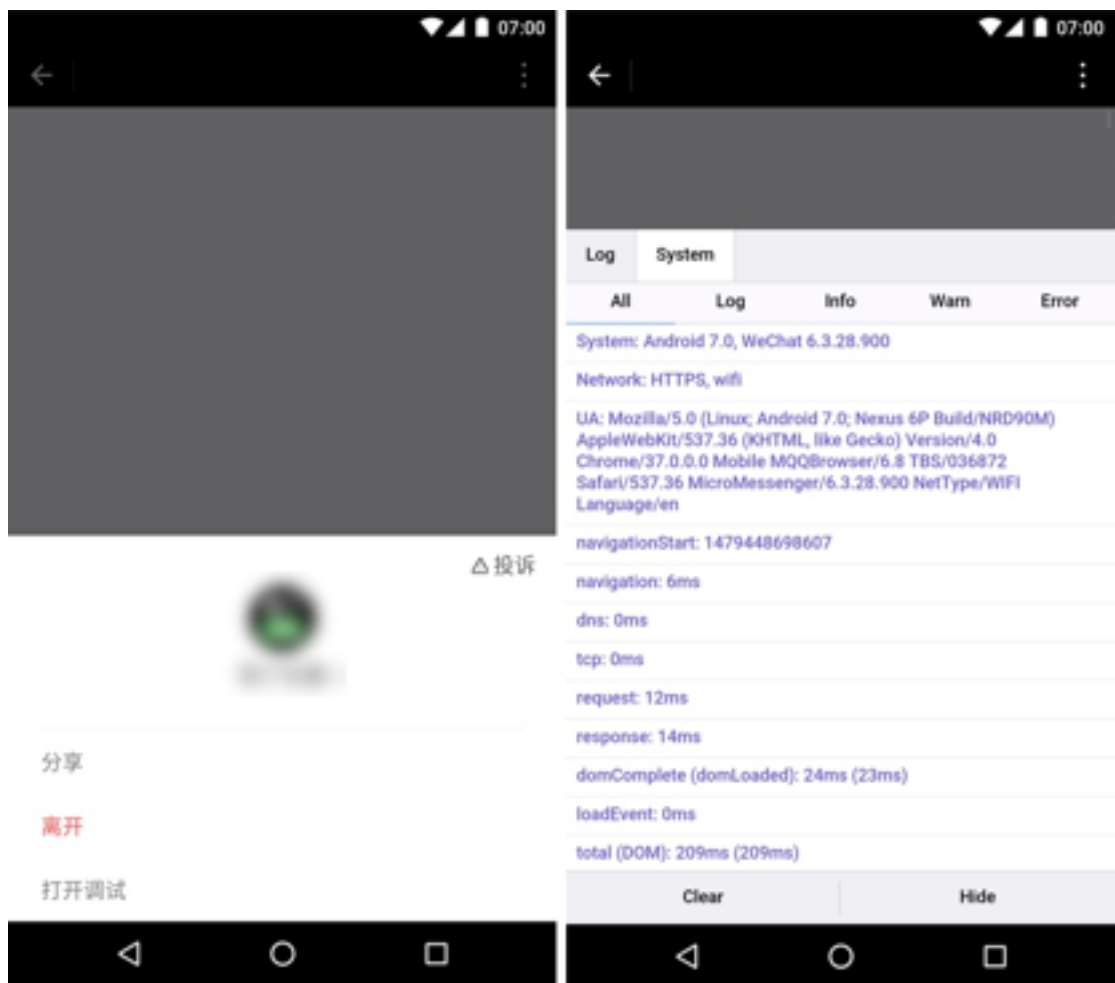


图 5-4-8 在手机上打开调试控制台

当小程序开发完毕之后，开发者可以点击「上传」按钮，将新版本代码上传到微信服务器。

服务器收到代码后，管理员就可以通过公众平台，将新版本提交审核。

以上就是知晓程序为大家带来的开发者工具使用及开发调试指南，希望能帮助小程序开发者迅速熟悉和上手小程序开发工具。

小程序的审核与发布

微信小程序什么时候会正式发布？也许连微信自己也不知道。

但是，我们知道的是，如果想成为第一批公开可用的小程序，就得赶紧将小程序提交审核。那是因为：

●先审核、先发布：

小程序虽然还不能公开使用，但可以提交官方审核。等到正式发布时，就不用排队候审了；

●很可能审核不过：

微信对小程序的审核非常细致，所以，你的小程序很可能不知不觉踩到红线而审核不通过。

如何让小程序快速安全地过审呢？现在，知晓程序（微信号 [zxcx0101](#)）就为你带来从提交代码到发布的全过程，帮你的小程序轻松通过审核。

此外，文末将会为大家带来一个小小彩蛋，内含为小程序添加体验者的方法。

第一步：上传代码

当小程序开发完成之后，开发者需要做的是将 **release** 版本代码上传至小程序在公众平台的后台。

1. 打开微信开发者工具，使用项目管理员的身份登录；
2. 打开需要提交新版本的项目；
3. 点击左侧栏「项目」；

4. 点击「最后上传时间」右侧的「上传按钮」；
5. 管理员扫码确认身份；
6. 填写版本号、项目备注等内容；
7. 确认无误后，点击「上传」。

上传确认

上传后，可以在公众平台查看本版本

版本号

合理设置版本号便于管理，只能输入字母、数字、. 如 v1.0.0

项目备注

可以备注项目版本优化内容等，便于管理员识别

取消 上传

图 5-5-1 开发者工具中的上传确认界面

点击上传后，**release** 代码会上传到小程序后台的「开发版本」一项中。

需要注意的是，「开发版本」中只能有一个版本的代码，新的上传操作会替代之前已存储在「开发版本」中的代码。

第二步：进入审核页面

当 **release** 代码上传完毕之后，开发者就可以将代码提交审核了。

在此之前，我们要找到提交审核页面的入口：

1. 进入微信公众平台（<http://mp.weixin.qq.com>），并使用小程序帐户登录；
2. 使用管理员微信扫码验证；
3. 进入后台后，点击左侧「开发管理」。



图 5-5-2 公众平台「开发管理」入口

此时，我们就进入了开发管理页面。

下拉页面找到「开发版本」一项，就可以找到我们刚才在开发者工具中上传的代码。

点击左侧的「提交审核」，使用管理员微信扫码验证并确认声明后，小程序就进入了审核资料提交页面。

第三步：填写审核资料

审核资料提交表单共分为「绑定测试号」和「配置功能页面」两个部分。

提交审核

① 身份验证 — ② 审核信息

绑定测试号 必填，用户微信审核人员登录审核使用

测试账号

绑定测试微信号
该微信号将提供给微信审核人员审核微信小程序时登录使用，微信号需能够体验小程序的全部功能，请勿使用常用微信号扫描

配置功能页面 至少填写一项，填写正确的信息有利于用户快速搜索到你的小程序

功能页面 请选择

标题 0/32

所在类目 请选择

标签
标签用回车分开，填写与页面功能相关的标签，更容易被搜索

添加功能页面

提交审核

图 5-5-3 审核资料填写界面

1. 绑定测试号

对于「绑定测试号」一项，微信官方的说明是这样的：

该微信号将提供给微信审核人员审核微信小程序时登录使用，微信号需能够体验小程序的全部功能，请勿使用常用微信号扫描。

微信在小程序官方运营文档中也提到过，**开发者需要为审核人员提供可体验所有功能的测试帐户**（一个供测试使用的微信号），以便**审核**；同时，还需要在帐户体系中新增微信登录功能。

如果小程序不带有帐户体系，可以跳过此步骤。

2. 配置功能页面

接下来，我们需要「配置功能页面」。在这一步，可以为小程序添加索引，以帮助用户分类检索小程序的相应服务页面。

那么，如何填写功能页面呢？分两种情况：

- **如果你的小程序只实现一种功能（服务）**：你只需要选择主页，填好标题并写好标签，并在「所在类目」一项中填写小程序的服务范围。
- **如果你的小程序需要实现多种功能（服务）**：你需要为实现每一种服务的主页，分别进行信息填写和服务范围标注。值得注意的是，你在审核时登记的服务范围，同时也需要登记在公众平台中（可以登记多个）。

没看明白？这里举两个例子：

- **如果你要做一个只卖手机的小程序**，那么，用户进入首页就得看到一个卖手机的页面。审核前，你需要在公众平台为小程序登记「卖手机」的服务类目；审核时，「功能页面」填写首页（**index**），服务类目选择「卖手机」。
- **如果你除了卖手机，还要在小程序里卖书、卖家具**？那么，小程序首页就要引导用户去到卖手机、卖书和卖家具的页面。

审核前，你需要在公众平台为小程序登记「卖手机」、「卖书」和「卖家具」三种服务类目；审核时，需要将卖相应物品的页面设为功能页面并附上相应的服务类目。

知晓程序（微信号 `zxcx0101`）推测，微信要求为小程序每一种服务填写服务范围和标签，可能与小程序的检索与用户入口有关。

因此，开发者需要认真考虑每一个功能页面的配置，以免影响小程序的露出。

第四步：等待审核和结果处理

提交审核之后，建议小程序的管理员时常登录小程序后台，检查审核结果。

审核完成后，管理员可以通过小程序后台页面右上角的「小铃铛」图标，检查审核结果。



图 5-5-4 审核不通过的通知

我们发现，在公测阶段，小程序审核不通过的原因，更多是一些官方文档中没有明确指出的问题。例如一些设计纰漏、交互误导等。

实际上，小程序的审核标准比已经公开的文档要更加细致、更加不可捉摸。

好在，微信的审核结果会将问题一条条展示出来。开发者只需要理解审核不通过的原因，有针对性地对小程序代码或审核资料进行修改，就可以重新提交审核了。

如果审核通过，管理员就可以在小程序后台点击左侧的「开发管理」，在「审核版本」中找到审核通过的版本。

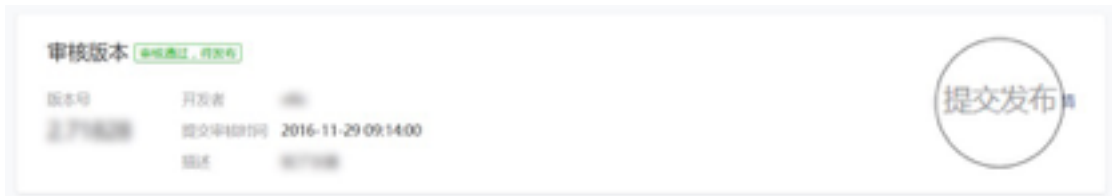


图 5-5-5 「提交发布」按钮位置

在这里，小程序管理员可以将审核通过的版本发布。但由于目前公测阶段尚不能公开发布小程序，目前的提交发布按钮依然是灰色的状态。

彩蛋：为小程序添加体验者

目前，小程序尚不能公开发布，也就是说，普通用户暂时还不能使用小程序。

但是，开发者有权向已经授权的体验者开放测试版权限，帮助开发者测试小程序在不同机型上是否能正确运行。

如何添加体验者呢？只需遵照以下几个步骤：

1. 管理员在公众平台的「用户身份」中选择「体验者」，点击「绑定」按钮，输入体验者的微信号；
2. 微信会向体验者发送绑定邀请，体验者接受绑定即可。



图 5-5-6 体验者管理页面

值得注意的是，如果你的小程序没有设置体验版，那么体验者也是无法正常体验的。

设置并分享体验版，其实也很简单：

1. 管理员在公众平台的「开发管理」找到「开发版本」一项，点击右侧的下拉菜单，选择「设为体验版」；
2. 设置完毕后，分享「开发版本」一项左侧的「体验版」按钮的二维码给体验者。

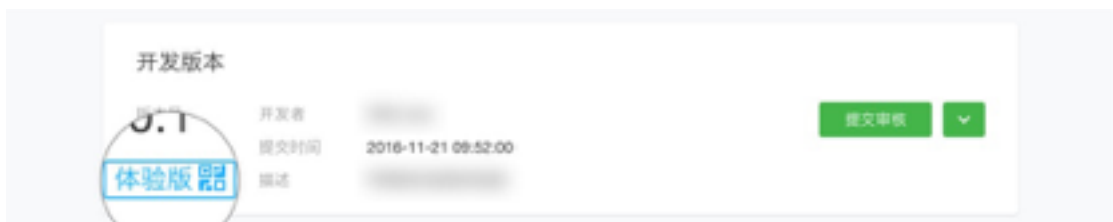


图 5-5-7 体验版二维码获取按钮

使用体验版前，你必须知道这些事情：

- 管理员可手动为小程序绑定体验者，每个小程序最多可以添加 40 位体验者。
- 在整个体验流程中（包括添加体验者、体验者测试小程序等）都不要要求体验版本过审。
- 所有微信用户（包括其他小程序的开发者或普通用户）都可以成为体验者。

管理员可以撤销体验版本，撤销体验版本的发布并不会令已经绑定的体验者资格失效。

以上就是知晓程序（微信号 [zxcx0101](#)）为大家带来的小程序审核和发布指南。希望本文可以帮助开发者快速将小程序提交审核，同时也祝各位开发者能顺利通过审核。

第六节 小程序官方文档解读

想顺利开发出小程序，微信提供的小程序官方文档你不得不读。

官方文档共包含三个部分，分别是开发文档、设计文档和运营文档。整个阅读和消化下来，差不多要花上一整天的时间。

如何能快速消化官方文档呢？知晓程序（微信号 **zxcx0101**）为大家准备了一份浓缩版的官网文档解读，只需要花一半时间，就能获得官方文档的全部重点。

开发文档解读

如果你想开发小程序，就得先学会一套微信特制的「开发语言」。

为了让大家上手这门开发语言，微信官方提供了一份十分详细的开发文档：

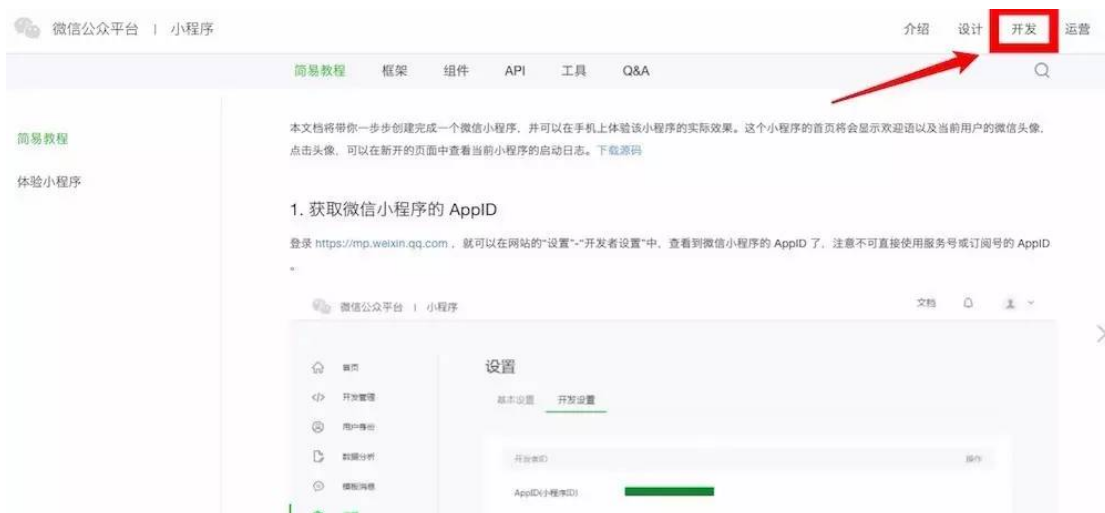


图 6-1-1 小程序开发文档入口

要是没时间通读，这里还有一份省时省力的替代品：

知晓程序（微信号 **zxcx0101**）准备了一份官方开发文档的超简浓缩版，不仅囊括了官方文档的所有重点信息，而且篇幅只有原文的三分之一。

只要看完这篇文章，你就能快速上手小程序开发。

一、 语言与文件

各位可能已经知道，微信小程序开发与其他平台的开发最大差异在于：微信使用的开发语言和文件很「特殊」。

小程序所使用的程序文件类型大致分为以下几种：

- **WXML** (WeiXin Mark Language, 微信标记语言)
- **WXSS** (WeiXin Style Sheet, 微信样式表)
- **JS** (JavaScript, 小程序的主体)

在语言方面，看上去小程序几乎重新定义了一套标准。但是实际上，它们与「前端三件套」——HTML、CSS 和 JavaScript——差不多。

下面，我们就来对比一下小程序开发语言和「前端三件套」有什么异同点：

- **HTML 与 WXML**：两者差异比较大，如果之前没有接触过 Android 开发，可能会觉得有些头疼。事实上，WXML 更像是 Android 开发中的界面 XML 描述文件，更适合于程序界面的构建；而 HTML 则倾向于文章的展示（这与 HTML 的历史有关），以及互联网页面的构建。
- **WXSS 与 CSS**：两者在语言上几乎没有差别，可以直接通用。
- **JS 文件**：小程序的 JS 文件与前端开发使用的 JS 几乎没有区别，小程序的 JS 新增了微信的一些 API 接口，并去除了一些不必要的功能（如 DOM）。

在语言上，小程序完全向学习成本最低的前端开发看齐，但这不代表所有前端开发者都能直接无缝迁移。

如果你是从前端开发转向小程序，就需要注意这两个点：

- **HTML 与 WXML** 两种文件的构建思想差异较大，如果之前只接触过前端开发，需要一点时间才能适应 WXML 的编写方法。

- 虽然小程序使用的是前端语言，但不代表可以继续沿用前端的开发思想进行开发（类似前端向 `Node.js` 发展）。小程序对前端开发的要求从「构建界面」升级成「开发完整应用」，前端开发依然需要在意识上进行转变。

二、 界面构建

1. 基本逻辑

`WXML` 和 `WXSS` 两种文件是小程序界面元素声明及样式描述文件。`WXML` 最大的特点是以视图（`view`）的方式串联界面元素，并通过程序逻辑（`AppService`），将信息更新实时传递至视图层。

`view` 类似于 `HTML` 中的 `div` 元素。在构建的时候，`view` 可以被多级嵌套，`view` 内可以放置任意视觉元素。

需要注意的是，元素一旦超出屏幕之外，用户是无法再看到的，这与 `HTML` 有较大不同。举一个例子，将手机屏幕想像成一个舞台，在舞台之外的演员是无法被观众看到的。

小程序有专门用于滚动的视图，如果希望界面是一个可以自由滚动的界面（例如列表等），可以使用 `scroll-view` 视图，在 `WXSS` 中将其大小调整为整个屏幕，并设置 `scroll-y`（上下滚动）或 `scroll-x`（左右滚动）为 `true`。

```
<scroll-view scroll-y="true">
  <view>Hello World</view>
</scroll-view>
```

小程序中不能直接使用 DOM 控制 WXML 元素。如果需要数据进行更新，需要使用 WXML 提供的数据绑定及元素渲染方法。

还有一点需要注意的是：小程序的栅格排版系统使用的是 Flex 布局，它是 W3C 在 2009 年提出的一种排版标准。

2. 绑定数据

对于单个字段，开发者可以使用数据绑定的方法进行信息更新。绑定的数据除了在加载的时候可以更新，也可以在 JS 主程序中以函数形式进行更新，更新同样可以反映到界面上被绑定的数据中。

3. 条件渲染与列表（循环）渲染

条件渲染适合带有意外情况提示的页面（如无法加载列表或详情时做出提示等等）。它的渲染带有触发条件，即符合条件时渲染这个页面，否则忽略或渲染另一段代码。

```
<view wx:if="{{ boyname == 'Tom' }}">Hello, Tom!</view>
```

```
<view wx:elif="{{ girlname == 'Anna' }}">Hello, Anna!</view>
```

```
<view wx:else>You neither Tom nor Anna.</view>
```

两个花括号所包含的判断条件中的变量于主程序 JS 代码中的 data 中声明。

```
Page({
  data: {
    boyname: 'Tom',
    girlname: 'Anna'
  }
})
```

若需要在界面中构建一个列表，可以使用 WXML 中的循环渲染，将同一元素渲染代码进行集合。循环的数据可以通过数组的方式写入 data 中供 WXML 访问。

```
<view wx:for="{{array}}">
  {{index}}: {{item.message}}
</view>
Page({
  data: {
    array: [{
      message: 'foo',
    }, {
      message: 'bar'
    }]
  }
})
```

渲染完毕后，渲染判断条件的变动可以影响界面变动。

4. 模板与引用

WXML 支持使用模板与引用减少代码体积。

模板是在 WXML 代码中对相同的代码进行复用的方式。

可以将多个模板写入至同一文件，并使用 `import` 在其他文件中引用。

```
<template name="greetings">
  <view>
    <text>Hello, {{name}}</text>
  </view>
</template>
```



```
//some other codes
<template is="greetings" data="{{...name}}" />
```

如果需要整个页面引用，需要使用到 `include`。

```
<include src="header.wxml"/>
  <view> body </view>
<include src="footer.wxml"/>
```

5. 样式

通过 `WXSS` 样式表，开发者可以定义 `WXML` 中的元素样式。

`WXSS` 与 `CSS` 代码一样，可以直接使用选择器选择元素。在 `WXML` 中也可以直接定义元素的 `id` 和 `class` 以便于在 `WXSS` 文件中进行样式定义。

6. 用户操作与事件响应

由于微信使用的不是 `HTML`，所以也不能通过添加超链接（`a` 元素）的方式来监测用户的点击事件。对于需要监听点击事件的元素，应该在 `WXML` 中使用 `bindtap` 属性或 `catchtap` 属性进行绑定。

```
<view bindtap="tapName"> Click me! </view>
```

除了点击一次，微信也提供按住、开始触摸、松手等事件响应。

在 `WXML` 中绑定好一个事件之后，就能在主程序 `JS` 中使用。

```
Page({
  tapName: function(somethingHere){
    // 执行这部分代码
  }
})
```

其他 `API` 中也有其他相应的事件，这些事件可以在微信小程序的官方文档中查阅到。

当需要在小程序的页面间进行跳转时，应该使用 `wx.navigateTo()` 方式。

```
tapName: function() {  
  wx.navigateTo({  
    url: '../logs/logs'  
  })  
}
```

需要注意的是，有关于页面层级跳转，微信将层级跳转限制到了五层。在开发时需要注意是否超过了相应限制。

三、 网络访问

小程序支持三种请求方式。一种是直接的 **HTTP** 连接请求，请求后直接返回结果，连接结束。另一种是 **Socket** 持续性连接，当一方主动关闭连接时，连接结束。

除了以上两种收发纯文本的连接方式，微信还提供了一个文件收发接口。小程序中录制的语音以及选择的照片都需要这个方式来进行上传。

通过小程序访问网络需要服务器端必须支持 **HTTPS** 安全连接，且端口号必须为 **443**。同时，小程序只能访问开发者在登记小程序时所设定的服务器地址。

四、 多媒体与存储

若需在小程序中播放多媒体（包括音视频）或进行数据存储，不能使用 **HTML 5** 中所提供的标准，必须使用微信提供的小程序多媒体播放控制接口及存储接口等。

有关于声音的接口有音频播放与音乐播放两种接口。音频播放提供了播放、暂停和停止播放三种接口，不提供跳转至某个播放时间点的功能，也不能获取目前的播放进度。音乐播放接口提供除以上的基础播放控制外的音乐状态检查和监听等功能。

小程序提供照片和视频数据交换接口。通过这个接口，小程序可以访问用户选定或拍摄的照片与视频。

通过音频录制和视频照片接口获得的多媒体信息是临时的，需要通过小程序存储文件接口对文件进行永久保存。

对于文本数据，小程序也提供了存储这类数据的接口。从诸如 **Android** 或其他 **app** 平台转向的开发者需要注意的是，小程序不提供数据库式的本地数据保存形式，而是通过「字段 - 值」的一对一形式进行保存。

五、 硬件相关

小程序依托于微信，提供许多与硬件有关的 **API**。以下数据，小程序可以通过 **API** 获取到。

- 系统相关信息（包括网络状态、设备型号、窗口尺寸等）
- 重力感应数据

- 罗盘数据

通过以上 **API**，应该可以轻松写出「摇一摇」等互动性页面。但需要注意：由于这些数据只能主动获取，而不能通过这些数值变化的回调实时获取。

六、 推送服务

小程序提供有推送服务，可以随时向用户发送必要的通知。但请注意，推送服务只能用于通知提醒，不能用于群发。

小程序中，推送服务叫做「模板消息」（之前有开发过服务号的开发者应该比较熟悉）。开发者需要在微信小程序后台登记新的模板推送消息（比如：购买成功通知等）并审核通过后，才能在小程序中使用模板消息推送服务，具体审核标准建议参考相应文档。

模板消息审核通过后，开发者需要先向微信服务器获取 **Access Token**，随后将该值、模板编号和模板中的动态变量（比如：订单号、价格等）提交给微信，由微信向用户推送通知。

七、 用户信息与微信支付

小程序可以在用户同意的前提下获取到用户的信息。小程序首先要通过微信登录的接口，让用户授权登录。之后，小程序就可以展示并使用用户信息。使用微信登录的时候需要注意，消息需要经过签名确认其完整性之后，方能保证数据未经篡改。

小程序中也可以使用微信支付。需要注意的是在发送支付请求时，需要在发送的消息中添加签名，以确认消息完整性。

以上便是知晓程序为大家整理的官方开发文档「重点信息」。

不过，想要熟悉小程序，光看是不行的，还需要动手去做。

如果开发过程中遇到问题，最好的办法还是查阅微信官方的小程序开发文档，相信在文档中，你能找到绝大部分问题的解决办法。

设计文档解读

从微信、公众账号、到微信支付，再到小程序，微信正逐渐将自己从一个「即时通讯工具」变成一个「操作系统」。但特殊的是，微信这个跨平台操作系统需要同时兼顾 iOS 及 Android 两套 UI 标准。

如何保证小程序的开发兼顾两种平台的界面风格，并与微信本身的体验保持统一呢？

阅读微信官方的设计文档是最有效的方式，里面提供了大量的用户界面实例，开发者只需严格依据这份设计文档，就可以做出兼具美观、实用的 UI 与交互模式。

为了帮助大家更快地理解设计文档，知晓程序(微信号 [zxcx0101](#))专门准备了这份文档解读，不仅将设计文档的篇幅浓缩到二分之一，还在文末增加了小程序 UI 设计与传统的移动应用设计和网页设计的差异比较。

现在，只需花一半的时间看这篇文章，你就能掌握设计文档的所有要点。

一、 基本规则

微信小程序设计文档将所有的设计范例分为了四个部分，分别是：

- 友好礼貌
- 清晰明确
- 便捷优雅

- 统一稳定

除此之外，微信官方还提供了一些便于开发者使用的组件。

设计文档也针对这部分组件进行了详细说明，包括每个组件的适用范围。同时，还提供了 Sketch 与 Photoshop 的设计模板，便于 UI 设计师直接使用。

知晓程序（微信号 zxcx0101）建议：没有太多设计经验的开发者请尽可能使用微信提供的组件样式，这样可以在快速开发的前提下，保证小程序的用户体验。

1. 友好礼貌

微信官方希望小程序专注解决用户当前的问题，做一个「管家」，而非是在用户处理问题时候，一直制造干扰的「推销员」。

首先，每一个页面都应该聚焦于某个重点功能，不能与该功能无关。

例如，搜索页面的功能应该是「搜索」，与搜索无关的功能或内容(如 banner 广告以及诱导搜索关键词推广等)应该尽可能去除掉。

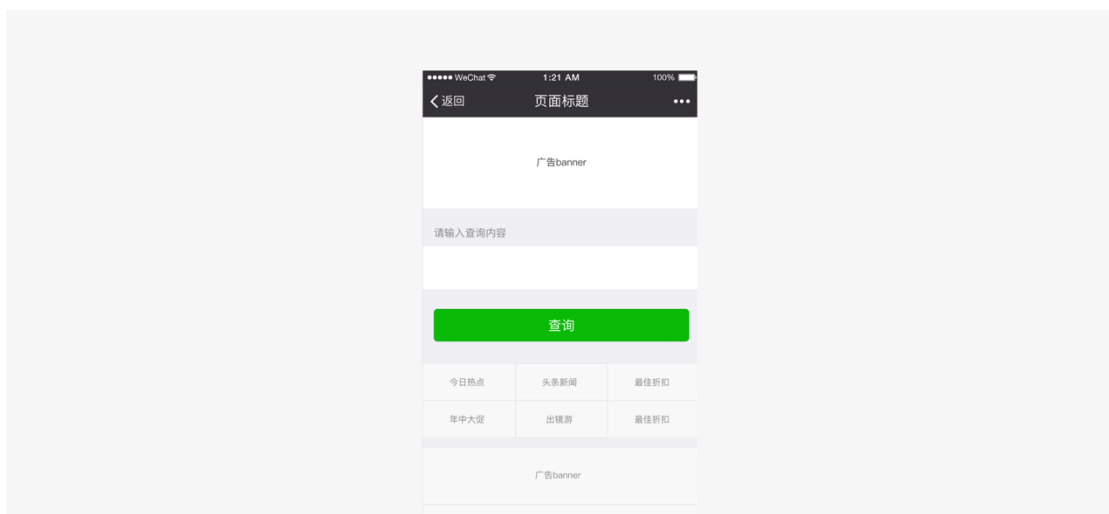


图 6-2-1 官方错误示例，在搜索页中添加无用信息

其次，页面的导航应该按照用户的预期进行。

例如，用户跳转进入一个页面时，小程序不应该弹出与之无关的广告，因为广告不符合用户进入页面的预期。



图 6-2-2 官方错误示例，进入页面弹出与功能无关广告

知晓程序（微信号 **zxcx0101**）建议：小程序设计时应该以快速解决用户问题为目标，并为其提供明确且符合预期的步骤导航。而这与张小龙在演讲中提到的「用完即走」的概念是吻合的。

2. 清晰明确

用户在操作小程序时，小程序应该为用户明示当前状态，同时应尽力减少用户在操作上的限制和等待时间。

微信官方已经为小程序提供了全局导航栏，包括导航区（返回按钮）、标题区和操作区三组。

如果开发者有需要，可以在小程序的首页中使用页面内导航，包括顶部 **Tab** 样式和底部标签样式。每种样式至少需要两个标签，最多不能超过五个标签，而微信官方给出建议是最多不要超过四个。

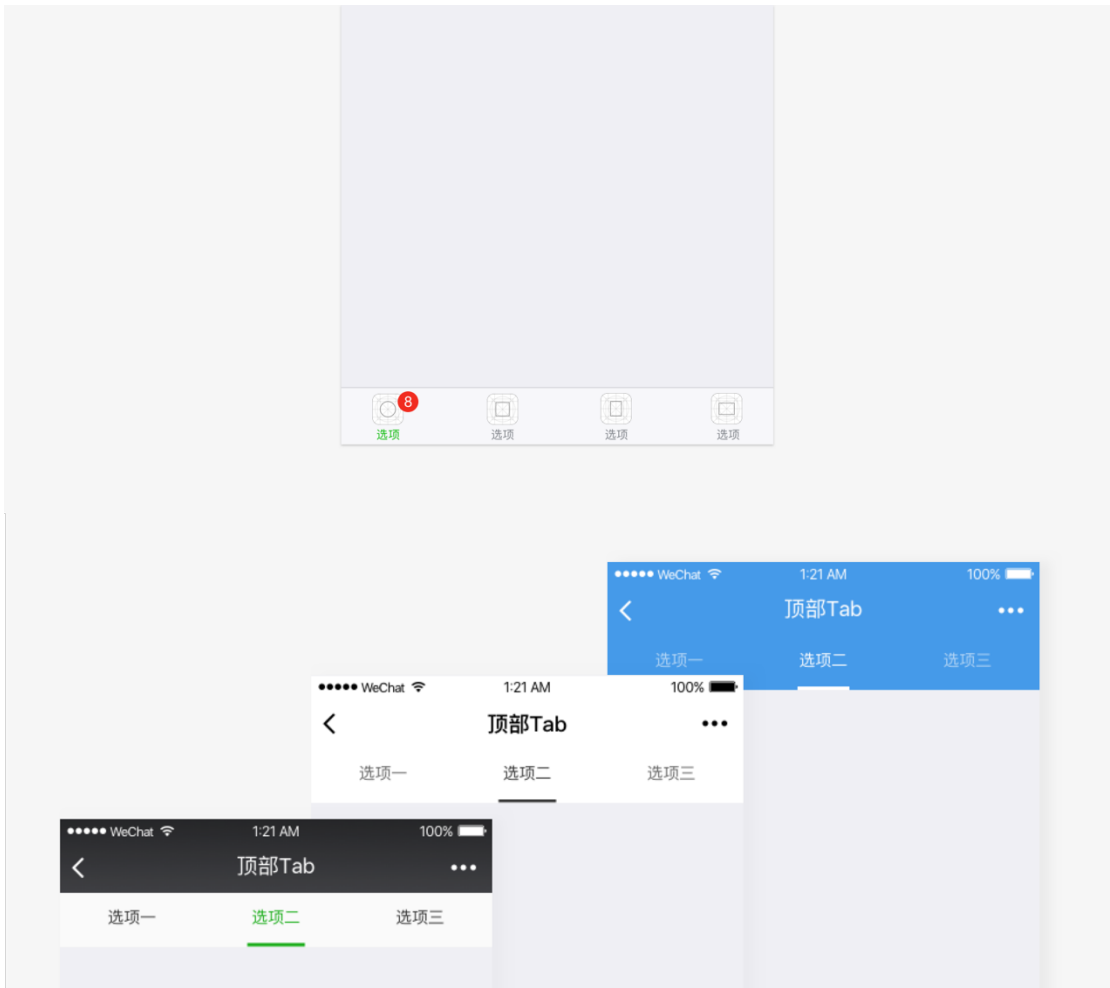


图 6-2-3 官方错误示例，在搜索页中添加无用信息

开发者可以定义导航栏和页面内导航的风格颜色。在定义颜色时，开发者需要注意元素辨识度，不能出现刺眼、按钮与文字可视性差的情况。

小程序中的加载反馈细节也是影响整体体验的重要部分。微信提供了一个下拉刷新组件，开发者只需要按照要求套用即可。

同时，页面内的加载提示推荐使用局部加载反馈而非全局加载反馈。加载时间较长时，建议提供进度条以减缓用户等待的焦灼感。

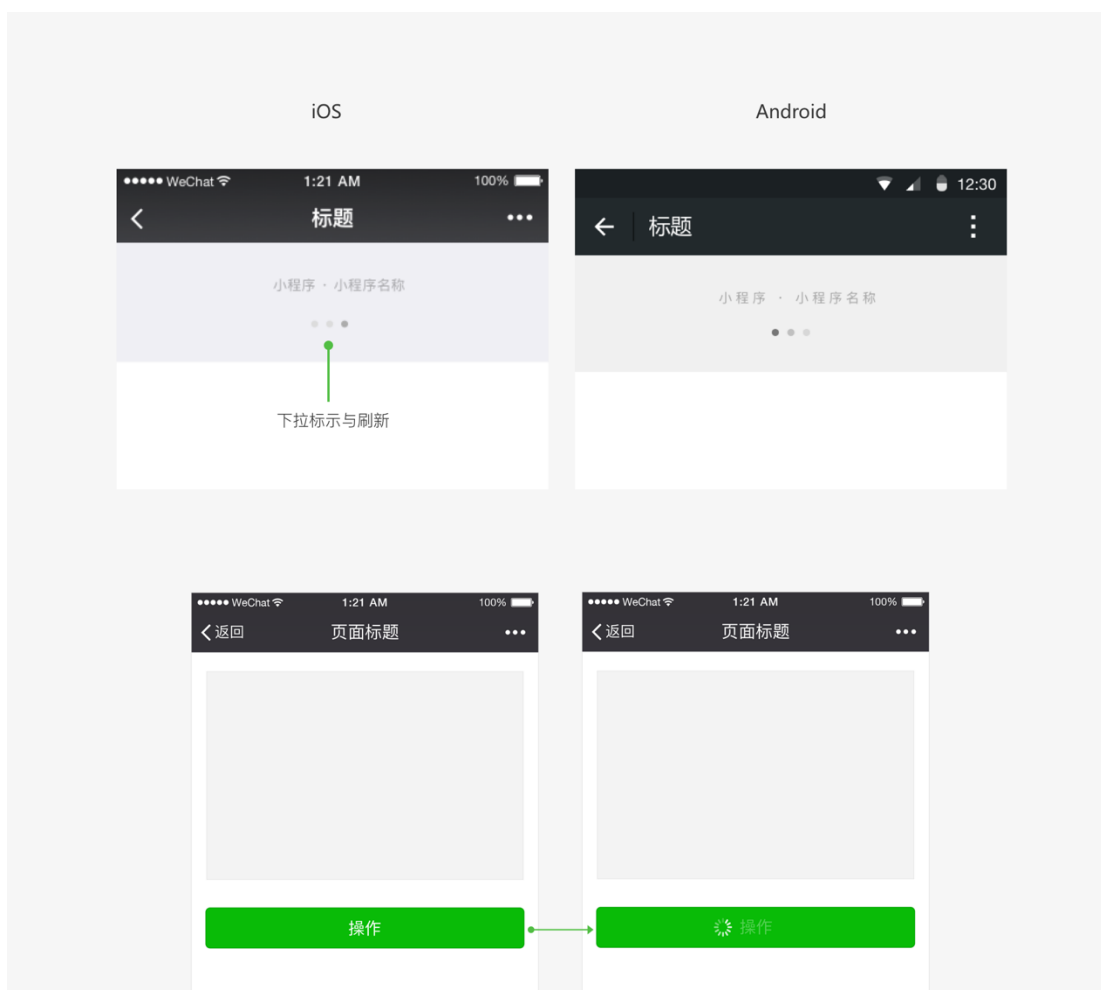


图 6-2-4 官方提供的底部标签和顶部 Tab 样式

微信小程序提供三个结果提示方式，提示效果从轻至强的分别为小弹窗提示（toast）、模态框提示（modal）和单独的成功结果页面。开发者需要根据实际情况，对不同程度的提示进行合理利用。

知晓程序（微信号 **zxcx0101**）建议：操作反馈时用 toast，询问行为时用 modal，表单提交后用页面。

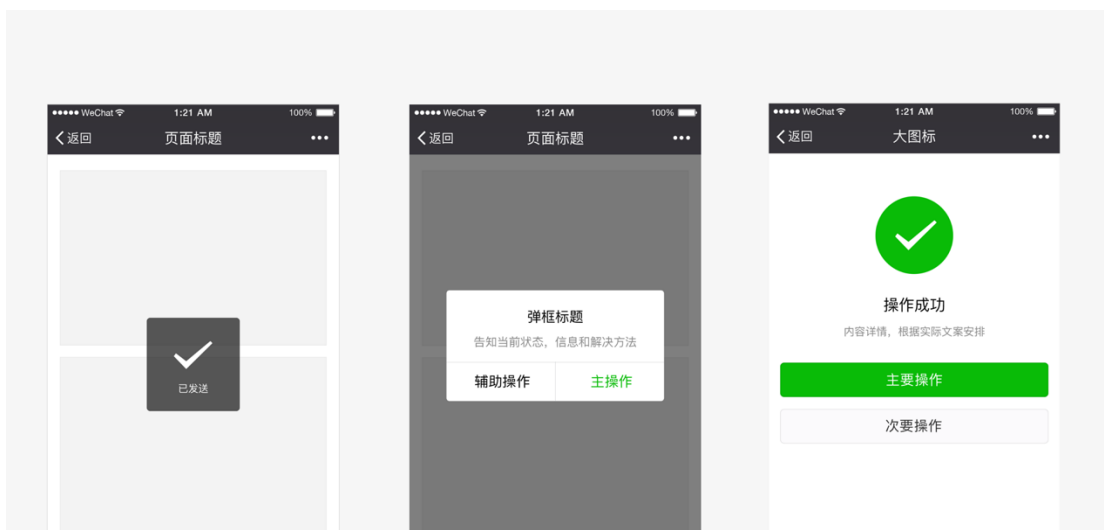


图 6-2-5 下拉刷新样式和局部加载示例

这份设计文档特别强调了异常情况的设计。在小程序中，异常情况需要明确告知用户出现了问题，同时明确告知用户问题出在哪里、应该如何解决。

文档对表单中填写出错情况的提示样式进行了规定。

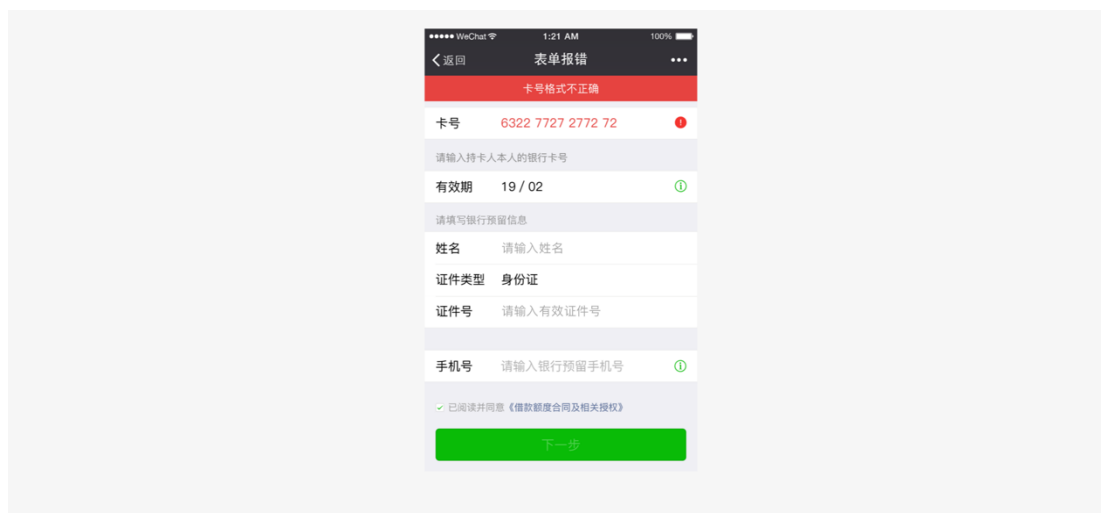


图 6-2-6 表单错误提示示例

如果表单中出现错误，小程序应该在顶部弹出提示，并在错误项目的右侧提供错误 icon，以使用户定位问题。

3. 便捷优雅

微信官方希望小程序可以敏捷地解决问题。

比如，当用户在进行输入时，可以通过联想、API 接口以及其他方式（例如扫描银行卡等），帮助用户快速准确地填充输入内容。

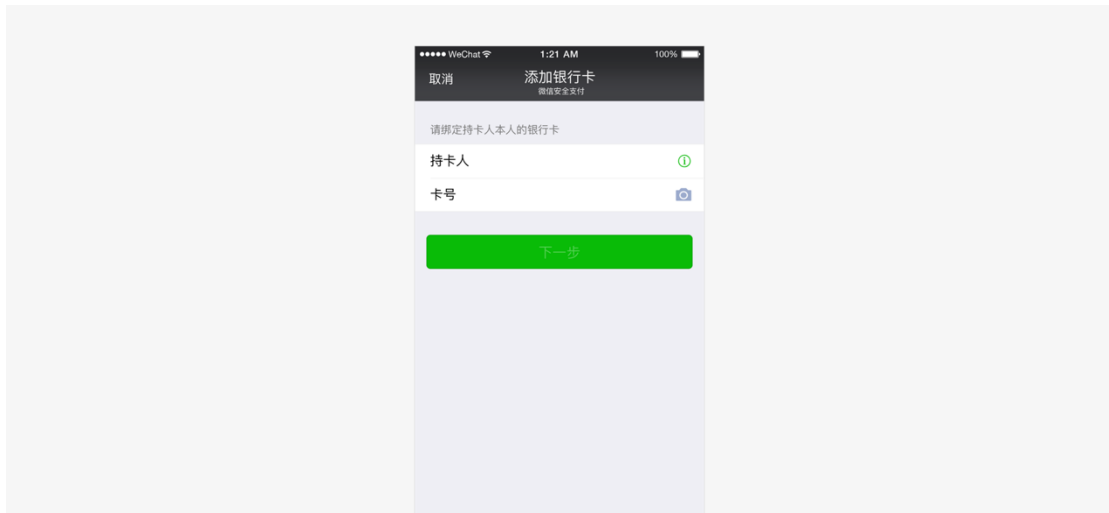


图 6-2-7 扫描银行卡快速填充内容

在设计时，应该确保用户的误操作概率更低。

微信官方在文档中提到，可点击元素应该要保证足够大，以使用户能够有明确的点击反馈。但知晓程序（微信号 zxcx0101）提醒，文案及 icon 不让用户困惑也是减少误操作的重要方式之一。

4. 统一稳定

在设计文档中，有关于统一稳定的说法更倾向于「统一」。即小程序在整体上应该要为用户提供整齐划一的功能，避免同一种视觉元素中在不同页面中有不同的样式。

这样的原则，有助于保证用户的认知稳定性。设计师在设计的时候，应尽可能避免一个小程序中多种元素风格差异较大的情况。这样

既不能保证小程序的视觉统一性，同时也对用户的认知稳定造成严重破坏。

知晓程序（微信号 **zxcx0101**）建议：在开发过程中，可以将相同的组件的样式规定于一个全局 **WXSS** 文件中，以便开发时快速复用并保证视觉统一。

二、 差异比较

如果网页设计师或移动应用 **UI** 设计师转向开发小程序的话，了解小程序与之前设计领域的异同是很有必要的。

我们以这份设计文档为基础，讲解小程序 **UI** 设计与传统的移动应用设计和网页设计的差异。

1. 如果你是网页设计师

对于网页设计师来说，小程序界面设计最大的挑战在于**以往桌面为先的思维需要转向移动为先**，因为作为一款依赖手机运行的应用，微信小程序需要聚焦于移动使用。

移动端与桌面端的区别在于：

- **屏幕尺寸与比例**。移动设备的屏幕一般会比桌面设备的屏幕更小，同时屏幕比例从横向变成了纵向。
- **输入设备**。桌面设备拥有鼠标和键盘，而移动设备上，用于信息输入的几乎只有手指。
- **网络特性**。桌面端不需要考虑网络问题，但在移动设备上，用户使用的网络可能会有流量限制或速率限制。

因此，知晓程序（微信号 [zxcx0101](#)）要提醒网页设计师，在设计小程序界面的时候需要注意这些差异：

- **多花点功夫在排版上。**在移动设备上使用程序，不像是在电脑上使用那么轻松自在。为小程序的元素和文字进行排版时，这些元素和文字应适当放大，同时加大间距。
- **避免花哨元素。**过于花哨的元素在小屏幕上容易「吸」走用户的注意力，不符合设计文档中的规范，同时会增加用户的网络流量和载入时间。
- **注意导航系统设计。**小程序没有「面包屑」（导航指示器），在设计的时候需要确保用户不会在页面中迷失，同时需要保证用户的下一步符合其预期。
- **多测试。**在转换至小程序的初期，强烈建议将设计原型稿放到真机进行测试和评估。

2. 如果你是移动 UI 设计师

对于 UI 设计师来说，小程序带来了一个好消息和一个坏消息。

好消息：移动 UI 中的设计思维和范式，绝大部分都能用在小程序设计上，且设计师不需要为 iOS 与 Android 分别设计界面，只需将有差异的部分进行分情况讨论即可。

坏消息：微信原生提供的控件较为有限。微信中只提供了多种按钮、Toast、提示 icon、开关、多选框、复选框和滑块等几种控件。其他的控件需要自己根据实际风格进行开发和设计。

此外，微信小程序的风格与 iOS HIG 和 Material Design 两种设

设计范式差异都较大。在设计时，需要多加注意设计文档中的说明和范例，以便设计出符合标准的小程序界面。

对微信小程序设计文档的解读就到这里为止了。希望通过这份解读文档，能帮助开发者轻松开发出具有优良体验的小程序。

运营文档解读

在之前的两篇文章中，知晓程序（微信号 [zxcx0101](#)）分别对小程序的官方开发文档和设计文档进行了浓缩和解读。

今天，知晓程序（微信号 [zxcx0101](#)）要解读是官方运营文档，包括小程序的「运营规范」和「常见拒绝情形」。

解读后全文篇幅不到原文档的三分之一，希望能帮小程序开发者节省阅读文档的时间，同时，避开小程序开发中的大坑。

一、 你的小程序能过审吗？

想做一个微信小程序？有想法是很好，但是在动手之前，需要注意你的想法适不适合做成微信小程序。

以下是微信官方禁止的小程序类别表：

- 非法、违反公序良俗等
- 功能过于简单
- 以传播谣言为目的
- 测试抽签（星座血缘等等）
- 以分级推销为目的
- 微信已有的功能（比如漂流瓶等）
- 恐怖性质较强（会给用户带来恐惧感）
- 要求用户越狱/root 后才能使用

除此之外，不在小程序服务范围的类别也无法通过审核。知晓程序（微信号 [zxcx0101](#)）在此特意提醒大家：**在开发之前，一定要检**

查小程序的分类是否能够通过审核。

关注知晓程序（微信号 **zxcx0101**），回复「服务范围」即可获得小程序服务范围的高清表格。

二、 设计注意事项

想法有了，接下来就是设计小程序的功能和界面。

在设计的过程中，除了界面需要遵循官方的设计文档外，功能设计上也有许多需要注意的事情。

关注知晓程序（微信号 **zxcx0101**），回复「设计文档解读」即可获得小程序官方设计文档的浓缩版解读文章。

1. 核心功能

微信官方希望小程序能以最快的速度解决用户发出的任务。因此，在功能设计上，小程序必须以解决问题为核心。

微信团队规定，小程序必须将核心功能前置，即用户打开小程序后，进入核心功能不得超过两次点击。

此外，每个页面都有唯一一个功能，在页面设计的过程中，页面的跳转必须符合用户的预期。

如果小程序页面中含有太多与主要功能无关的功能，就比较容易
被拒绝。这点在设计中值得注意。

2. 广告

知晓程序（微信号 **zxcx0101**）在设计文档中提到对广告和推广的限制。

关注知晓程序（微信号 **zxcx0101**），回复「设计文档解读」即可获得小程序官方设计文档的浓缩版解读文章。

而在微信官方的运营文档中，广告可以说是被严格限制的。

如果要在小程序中做推广，不可对用户造成严重的骚扰：首先不要投放全屏广告，也不能让广告在页面的主要部分被凸显。

在设计阶段，务必严格按照官方设计文档来进行。如果还想知道小程序中怎样的广告是可以接受的，那么微信官方的朋友圈广告和公众号末尾广告都可以作为参考。

3. 信息收集与处理

微信官方对使用小程序收集和处理信息的行为，进行了严格规定。

首先，在小程序与服务器交换数据时，所有数据都必须通过加密的 **HTTPS 443** 进行连接。

在收集数据的时候，需要明确告知用户数据收集的内容和目的，由用户决定是否提交数据。

如果小程序有外部分享功能（如分享至微博等），也需要明确告知用户小程序正在分享、分享的内容是什么等等。

在设计的过程中，知晓程序（微信号 **zxcx0101**）建议在用于提交数据的元素（例如文本框等）中，需要对用户提交的数据进行明显的提示，不要出现含糊不清的情况。

如果小程序带有持续性位置数据获取功能，也需要在设计中有明显提示。例如，增加「正在获取位置信息」的文案，或是在小程序中展示带有相同意义的 **icon**。

如果有能力，开发者可以与用户签订 **ToS** (**Term of Services**, 服务协议) 以及隐私声明等方式，来提示用户数据使用的目的等。

4. (被禁止的) 诱导

小程序在功能设计的时候，**万不可诱导用户先关注某些公众号或小程序，才能使用功能。**

目前，许多 **app** (特别是在 **Android**) 有一种「自定义组件」的方式，即下载安装主程序(本体)之后，用户可以依照自己的需要下载安装其他组件，以完成功能。

在 **iOS** 和 **Android** 上，这种方式有助于压缩应用体积并提高运行效率。但在小程序中，这种行为涉嫌诱导关注。所以知晓程序(微信号 **zxcx0101**) 建议：用一个独立小程序直接完成所有任务。

诸如「欢迎扫码关注 **xxx** 公众号」「欢迎扫码下载 **xxx** 应用」「关注或下载 **xxx** 方可查看」，以及「不关注 **xxx** 不是中国人」这种事情还是少干为妙。

5. 运维功能

在功能设计的过程中，需要注意添加一些运维性功能。

所有的小程序都需要有应用内反馈机制，以使用户在应用内进行问题反馈。在设计和规划时，这个功能就应该被考虑在内。

对于社区及 **UGC** 小程序，小程序内也需要提供举报(报告疑虑)机制，以便管理员能够即时对违反 **ToS** 及相关法律法规内容进行处理。

三、 开发注意事项

设计完小程序，总算可以步入开发阶段了。

微信官方提供的运营文档中，明确标明了**审核时需要提交一份技术说明文档**。目前尚不清楚是需要提交审核的时候提交，还是在审核人员要求时才需要提交。

所以，在小程序的开发过程中，开发人员需要注意记录每一个功能点，包括功能名称、触发位置、实现手段等等，以便于在审核时进行提交。对于功能较多的小程序，这部分的总结工作更需要留意。

对于包含用户中心功能，或需要用户登录才能使用某些功能的小程序，需要在技术说明文档中提供一份用于审核的用户名和密码，以便审核人员能够完整审核小程序。

知晓程序（微信号 **zxcx0101**）建议：在开发的过程中提供一个供审核人员使用的入口。

开发的最后，还需要检查小程序本身能否正常运行。如果审核时小程序出现严重的不正常情况，是不能通过审核的。

有多媒体播放功能的小程序还需检查是否有多媒体意外自动播放的情况，自动播放多媒体也是审核被拒绝的原因之一。

四、 审核之前，你需要注意

只有拿到了 **App ID** 的小程序才能进行调试，并提交审核。

在此之前，还需要完成小程序的资料填写等过程。

填写资料时，小程序的名称除了不能与现有的公众号重名之外，

还需要注意以下几个方面：

- 名称需要与小程序相关
- 不能直接起名为泛名称，类似「电话」、「短信」、「邮件」等
- 不能出现新广告法明令禁止或其他无关的词语
- 对于预留名称，需要提交额外资料进行审核

上传小程序 **Logo**（头像）时，图片需保持简洁、清晰、可辨识。同时，微信官方不建议在小程序 **Logo** 上加背景和边框等（类似一些 app 处理 iOS 应用图标的情况）。

此外，小程序简介需要与小程序本身有相关性，即介绍小程序的功能和作用。

知晓程序（微信号 zxcx0101）再次提醒：提交审核前，请以正常用户的身份再测试一次小程序。

五、 运营

如果通过了审核，那么，你的小程序就可以正式上线了，恭喜！但是，未来的运营中也有许多情况需要多加留意，以免小程序被勒令取消发布。

1. 内容运营

在日常的运营过程中，小程序的内容不能出现以下信息：

- 违法、违规
- 谣言、虚假不实的消息等

- 诱导性信息（诱导关注公众号、添加小程序）
- 造成骚扰的信息，或进行恶意营销
- 会对用户造成恐惧的信息

此外，如果你的小程序有 **UGC** 功能，则需要及时审核，以便尽早发现违规内容，避免小程序因为以上违规内容的传播而带来不必要的麻烦。

2. 信息安全

不能在未经用户同意的前提下，擅自转移或销毁用户数据。如果小程序的生命周期结束或被收购，那么，小程序的运营方需要为用户提供数据处理的解决方案。

不能私下买卖和泄露用户在小程序中提交的数据，未经用户同意对外展示的资料也不允许公开。

总之，在小程序中使用数据会受到严格的控制，开发者需要尊重用户对于数据的控制权并保障数据安全。

3. 模板消息

小程序具有模板消息推送能力。通过模板消息功能，小程序可以向用户推送重要通知。

需要注意的是，小程序中的模板消息只具有通知功能。而且，触发特定事件（如生成订单等）后，小程序才能向用户推送消息。

小程序主动发送模板消息，或诱导用户触发模板消息的行为，都违反了相关条例中的说明。

微信官方更倾向于将模板消息作为一种通知手段，而非一种推广

方式。如果小程序违规使用模板消息，微信团队会对小程序做出处罚，甚至会让小程序下架。

4. 服务可靠性

如果小程序的运行依赖运营方提供的服务器，那么运营方需要保证服务器的稳定。不稳定的服务器不仅会影响用户体验，而且可能会让小程序下架。

对于更新周期较长的小程序，知晓程序（微信号 **zxcx0101**）也建议运营方时常登录小程序公众号后台，以免小程序因长期不登录而被下架。

某些小程序可能会通过热更新的形式动态加载代码，以缩短小程序的更新周期。虽然热更新的内容不需要另行经过审核，但其中也不能出现任何违规行为。一旦热更新的内容被发现违反相关规定，那么小程序有可能会被勒令下架。

以上就是知晓程序（微信号 **zxcx0101**）对官方运营文档的全部解读，希望开发者仔细阅读文章内容，避免未来出现违规行为，而导致自己的小程序处罚下架。

对微信小程序官方文档的解读到此就完结了。关注知晓程序（微信号 **zxcx0101**），在后台回复「解读」，就能获取全部的解读文档：

- 开发文档解读：开发微信小程序，这篇文章就够了
- 设计文档解读：如何开发好看又好用的小程序？
- 运营文档解读：不想你的小程序被下架？收好这篇避坑指南

第七节 电商类小程序开发教程

看到这里，相信大家对小程序这个新事物，已经不陌生了。

如果想亲手做一个小程序，应该怎么开始呢？

知晓程序（微信号 **zxcx0101**）邀请了爱范儿的技术团队，为我们专门编写了一套电商类小程序开发教程。也许，你也能很快开发出自己的小程序。

如何做轮播图

前段时间，微信正式对外发布了小程序的公测，虽然暂时不支持对外发布，但也意味着有更多的开发者可以参与到小程序的开发中来。

作为第一批获得微信小程序内测资格的媒体，爱范儿（微信号 ifanr）的技术团队曾用一个上午时间，完成了旗下电商平台玩物志（微信号 coolbuy）的小程序 Demo，并第一时间分享了开发经验，引发广泛关注。

今天知晓程序（微信号 zxcx0101）就以玩物志的小程序为例，演示如何一步步开发一个完整的电商类小程序，并分享小程序开发过程中需要注意的点。

注：本系列文章的 Demo 采用微信小程序 2016.10.31 更新的 0.10.102800 版本，在实际操作中可能有出入。

玩物志小程序跟常见的电商平台类似，主要由首页、分类页、商品列表页、商品详情页、购物车、订单结算页和个人中心等几大模块组成。

知晓程序（微信号 zxcx0101）会分几期来对每个模块的关键点作详细说明。

在此过程中，也会穿插介绍一些常用简单内置组件的使用，最最后利用小程序的 `template` 模板，对自定义组件化开发的实现提供三种参考模式。

一、 准备开工

我们先新建一个文件夹 `coolbuy` 作为项目目录，里面包含：

- `common` 目录——用来放置项目公用的文件
- `pages` 目录——项目页面开发目录，里面的每个子目录代表一个独立页面，分别包含与目录同名的 `.wxml`（必须），`.wxss`，`.js`（必须）和 `.json` 文件
- `templates` 目录——用来放置自定义的组件
- `utils` 目录——用来放置工具类 `js` 文件
- `app.js`——小程序逻辑
- `app.json`——小程序公共设置
- `app.wxss`——小程序公共样式表

二、 轮播 banner 样式定义

接下来，是首页的实现。

首页在 `pages/index` 目录下，包括 `index.wxml`、`index.wxss`、`index.js` 三个文件。

首页的完整 Demo 图如下：

玩物志—新生活必需品

首页

最新

最热



新品上架



包治百病



潮表精选



数码周边



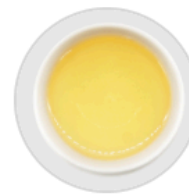
风骚小物



家居百货



文房四宝



味蕾培养

New Arrivals
新品上架



首页



分类



购物车



我的

图 7-1-1 首页最终效果

第一步，实现 banner 轮播图，来看代码：

```
/** index.wxml */
<swiper class="banner-swiper"
indicator-dots="{{ indicatorDots }}"
autoplay="{{ autoplay }}" interval="{{ interval }}"
duration="{{ duration }}">
  <swiper-item class="swiper-item"
wx:for="{{ banners }}">
    <navigator
url="../detail/detail?id={{ item.productId }}"
  <image src="{{ item.image }}"
class="slide-image"></image>
    </navigator>
  </swiper-item>
</swiper>
```

值得注意的是，`image` 组件会有一个默认的风格。

官方文档中说 `image` 组件默认宽度 300px、高度 225px，但是，根据开发者工具 v0.10.102800 版中的测试，`image` 组件的默认样式为：

```
image {
  width: 320px;
  height: 240px;
  display: inline-block;
  overflow: hidden;
}
```

这个值可能会在小程序以后的升级中进行修改。

知晓程序（微信号 `zxcx0101`）提醒：跟 `image` 组件类似，大多数小程序内置组件都会有自己的默认样式，初学者可以利用开发都工具中的 `Wxml Panel`，多了解一下这些组件的默认样式，可以避免很多由组件默认样式引起的问题。

我们要根据实际情况给它重新定义一个宽高值：

```
/** index.wxss */
.slide-image {
  width: 750rpx;
  height: 450rpx;
}
```

在这里使用了小程序特有的 `rpx` 单位，这个尺寸单位会放到第二期的页面布局中来讲。

另一个注意点是，不要将大图片放到项目目录中，因为小程序代码总大小不能超过 `1 MB`，而大图片会占用大量的空间，所以要尽量避免在项目目录中放置大图片。

如有需要，建议只在项目目录中放置少量 `icon` 类的小图片，其他大图片可以上传到自己的服务器或者网盘中，然后在 `src` 中设置图片的网络资源地址。

在手机测试或者正式上线之前，要注意是否有引用了本地大图片导致代码超出大小限制的问题。

三、 准备获取 banner 数据，并绑定到视图层

接下来看数据的绑定。

```
/** index.js */
Page({
  data: {
    banners: [], //初始化一个 banners 数组

    //以下四项设置 swiper 组件的参数值
    indicatorDots: true
    , autoplay: true, //开启自动切换

    interval: 5000, //自动切换时间间隔

    duration: 1000 //滑动动画时长
```

```
    }  
  });
```

在 Page 里的 data 会以 JSON 的形式由逻辑层传至视图层，然后在视图层中使用 Mustache 语法（双大括号）接收传入的数据，实现动态数据的数据绑定。

经过绑定后的数据变动可以在视图中实时反应，也可以通过开发者工具中的 AppData Panel 中查看和编辑页面中的所有 data 数据。

知晓程序（微信号 zxcx0101）提醒：当你不知道 data 数据是否被设置成功的时候，可以到 AppData Panel 来查看一下，免去了手动打 console.log 进行调试的麻烦。

实现动态数据绑定的另一个方式是使用 this.setData() 方法，this 在这里代表的是 Page，而 setData 是在 PageX.prototype 中的方法，所以 this.setData() 实际上访问的是 Page.prototype.setData()。

它可以用来新增和修改 this.data 里的值，同时还有一个重要的功能是将数据从逻辑层发送到视图层，从而实时改变页面的状态，保持逻辑层和视图层数据的一致性。

需要注意的是，如果使用 this.data.property = value，同样可以设置 this.data 的值，但是它不会将数据发送到视图层，无法改变页面的状态。

知晓程序（微信号 zxcx0101）提醒：一定要避免使用 **this.data.property = value** 的方式设置数据。

四、 从服务器获取 banner 数据

言归正传，现在我们的目的是要得到一个 `banners` 数组，然后利用 `wx:for` 将 `banners` 数组中的每一项渲染到 `swiper-item` 中。

我们可以在页面加载 (`onLoad`) 时，使用 `wx.request()` 发起一个 Ajax 请求，来获取到所需要的 banner 列表数据。

先来看一下 `wx.request(object)` 的使用方法。首先是 `object` 可以接受的参数值：

- `url`: 设置发送请求的地址。
- `method`: 请求方式，默认为 `GET`，有效值: `OPTIONS`, `GET`, `HEAD`, `POST`, `PUT`, `DELETE`, `TRACE`, `CONNECT`。知晓程序（微信号 `zxcx0101`）提醒，所有值都要使用大写。经过我们测试，若使用小写，在安卓中会导致无法发起请求，而在开发者工具和 `iOS` 中则正常。
- `data`: 设置请求的参数。如果 `method` 值为 `GET`，这个函数会将 `data` 转换成 `query` 字符串，拼接在 `url` 后面，若 `url` 本身带有同名的 `query` 参数，那么 `data` 中的参数会覆盖 `url` 中的 `query` 参数；

小程序执行 `wx.request(object)` 发送请求并收到 `success`（成功）的消息后，会自动执行请求成功返回的回调函数。

回调函数执行时，会自动收到类似这样的回调参数：

```
res = {data: '服务器返回的数据', errMsg: '错误信息'}
```

```
' , statusCode: HTTP 状态码}
```

特别注意，在开发者工具和 iOS 中，`res.statusCode` 的数据类型是一个数值，而在 Android 中的数据类型是一个字符串，所以在判断 `res.statusCode` 时不能直接使用 `===` 操作符，而应该用 `==` 操作符，或者是先对 `res.statusCode` 进行数据类型转换。

现在，就可以在 `onLoad` 中调用 `wx.request()` 发起请求，访问获取 banner 列表的服务器 API。

我们将各服务器 API 的访问地址都保存在 `API_URL` 对象，保存在 `utils/config.js` 中：

```
/** config.js */
const CONFIG = {
  API_URL: {
    BANNER_QUERY: http://ifanr.com/api/banner
  }
}
module.exports = CONFIG;
```

然后需要在 `index.js` 中使用 `require` 引入 `utils/config.js` 文件，引入路径为当前文件的相对路径：

```
/** index.js */
var CONFIG = require('../utils/config.js');
```

于是，我们就知道了在小程序中 `js` 模块化的方法，模块通过 `module.exports` 对外暴露接口，在需要使用这些模块的文件中，使用 `require(path)` 将公共代码引入。

发起 `wx.request` 请求：

```
/** index.js */
var CONFIG = require('../utils/config.js');
```

```
Page({
  data: {
```



```

    ...
  },
  onLoad: function () {
    var _this = this;

    wx.request({
      url: CONFIG.API_URL.BANNER_QUERY,
      method: 'GET',
      data: {
        limit: 6,
        img_size: 'small'
      },

      success: function (res) {
        if (res.statusCode == 200) {
          _this.setData({
            banners: res.data.objects
          });
        }
      }
    });
  }
});

```

在 `wx.request()` 的回调函数中判断 `res.statusCode` 是否为 200，若为 200 则代表请求成功，接着将数据保存到 `banners` 数组中。

我们在 `success` 回调函数中使用 `_this.setData({ banners: res.data.objects });` 设置 `banners` 的值。

知晓程序（微信号 `zxcx0101`）提醒：这里不能直接使用 `this.setData()`，因为 `success` 回调函数是一个闭包，我们要先在闭包外面定义 `var _this = this;`，再调用 `_this.setData()`，以保证闭包内能到正确的 `this` 值。

现在，在视图中我们就能看到完整的 `banner` 列表了。

本文介绍了小程序中内置组件、数据绑定和发送请求 **API** 的用法，提醒了初学者一些不容易注意到的坑，通过上面的介绍和实际代码演示，希望能帮助初学者更容易理解和使用小程序的其他组件和 **API**。

下一期，知晓程序(微信号 **zxcx0101**)将会介绍小程序中 **flex** 布局结合 **rpx** 的应用和 **template** 模板以及交互反馈类 **API** 的使用。

如何做商品列表

在上一期文章中，我们以爱范儿旗下的玩物志（微信号 `coolbuy`）电商小程序作为 **Demo**，介绍了首页 **banner** 部分的开发，以及微信小程序内置组件、数据绑定和发送请求 **API** 的用法。

本期，知晓程序依然以玩物志电商小程序为基础，为大家带来有关于列表加载的实现教程。

一、 为布局而生的 **Flex**

传统的布局方式一般都是基于 **div** 盒子模型，利用 **float**，**position**，**display** 来进行布局。

每个前端开发者对这些布局方式都非常熟悉。对一些特殊的布局来说，使用这些属性并不方便，由此还衍生出各种 **hack** 方案。直到 **CSS3** 提出了专为布局而生的解决方案，就是 **Flex** 布局。

但是，由于浏览器的兼容问题，大多数开发者都不敢将 **Flex** 布局直接应用到实际项目之中。在常规项目中，大量应用 **Flex** 布局还需要一个过程。

而对于 **WeChat Only** 的小程序来说，则完全不用担心有浏览器兼容的问题，它只要适配微信客户端即可。同时，由于小程序的布局不会非常复杂，所以，也不用过多地担心大量使用 **Flex** 引起的性能问题。

我们可以放心地在小程序中使用 **Flex** 来处理布局。

二、 如何更方便地使用 rpx

rpx 的全称是 responsive pixel，它是小程序自己定义的一个尺寸单位，可以根据当前设备的屏幕宽度进行自适应。

小程序中规定，所有的设备屏幕宽度都为 750rpx，根据设备屏幕实际宽度的不同，1rpx 所代表的实际像素值也不一样。

例如，在 iPhone 6 上，屏幕实际宽度为 375px，则 $750rpx = 375px$ ， $1rpx = 0.5px$ ；而在 iPhone 5 上，屏幕实际宽度为 320px，则 $750rpx = 320px$ ， $1rpx = 0.42px$ 。

其实，我们并不必关心每种设备屏幕下 1rpx 到底代表多少个像素，只要抓住「所有的设备屏幕宽度都为 750rpx」这个原则，就能很好地实现对任意设备屏幕大小的自适应布局。

知晓程序（微信号 zxcx0101）强烈建议用 iPhone 6 作为视觉稿的标准，即将视觉稿总宽度设成 750px。这样，开发者就能很方便地对相关的尺寸进行量取。

比如，在总宽度为 750px 的 iPhone 6 视觉稿中，量取一个图片的宽度为 200px，那么，这个图片的宽度即可设置为 200rpx。

简单一句话解释：量取多少就设置多少。

三、 Flex 配合 rpx 的使用案例

现在，我们就在小程序中使用 Flex 和 rpx 进行布局，体验一下这种解决方案的便利。

案例一：货架列表

来看看货架列表的效果图：



图 附-3-2-1 货架列表效果图

首先还是贴两段代码，以下是 WXML 的实现：

```
/** index.wxml */  
<view class="shelf-nav">  
  <view class="shelf-nav-item"  
wx:for="{{ shelfNavList }}">  
    <navigator  
url="../list/list?id={{ item.id }}">  
      <image src="{{ item.cover_image }}"  
class="shelf-nav-item__image"> </image>  
      <text>{{ item.name }}</text>  
    </navigator>  
  </view>  
</view>
```

然后 WXSS 的代码如下，我们在关键代码处已经给出简要注释：

```
/** index.wxss */  
  
.shelf-nav {  
  display: flex; //设置 display: flex; 将它变为  
flex 布局的元素  
  
  flex-wrap: wrap; //当子元素总宽度超过父元素宽度时换
```

行显示

```
padding: 30rpx;
}

.shelf-nav-item {
width: 25%; //因为每行显示 4 个货架，所以宽度设置为
25%;
margin-bottom: 20rpx;
text-align: center; //让它中间的图片 and 标题居中显示
}

.shelf-nav-item__image {
width: 130rpx; //在视觉稿中量取图片宽高为 130px
height: 130rpx; //于是设置 width 和 height 都设置
为 130rpx;

border-radius: 50%; //把图片设置成圆形

border: 1px solid #d9d9d9; //加上外边框
}
```

简单几步就完成了货架列表的布局，并且完美兼容各种大小的设备屏幕。

案例二：货架分类标题

再来一个例子，看一下商品列表货架分类的标题：

New Arrivals

新品上架



预售 | 「Usmile」U1001 电...
¥ 399.00



「Cabeau」一把长歪了的伞
¥ 299.00

图 附-3-2-2 分类货架效果图

如上图所示，我们需要做到的效果是货架标题居左，「查看更多」的图标居右并且垂直居中。

代码结构如下：

```
/** index.wxml */  
<view class="shelf-header">  
  <view class="shelf-title">  
    <text  
class="shelf-title-en">{{ shelf.english_name }}</  
text>  
    <text  
class="shelf-title-cn">{{ shelf.name }}</text>  
  </view>  
  <view class="shelf-more">  
    <navigator  
url="../list/list?id={{ shelf.id }}"  
class="ifanrin-more-icon ifanrin"> </navigator>
```

```
</view>
</view>
```

有经验的同学一眼就可以看出，要实现需求需要：

- 让 `.shelf-title` 向左浮动
- 让 `.shelf-more` 水平方向右对齐、垂直方向居中
- 最后还要给 `.shelf-header` `clearfix` 一下

那现在来看看 `Flex` 是怎么做的：

```
/** index.wxss */
.shelf-header {
  display: flex; //设置为 flex 布局的元素

  justify-content: space-between; //均匀排列每个元素，首、末两元素分别位于起点和终点
}
```

```
.shelf-more {
  align-self: center; //设置居中
}
```

只要区区几行代码，就能轻松使用 `Flex` 调整排版。

接下来，复习上一期的内容，发起一个请求获取数据，然后

`setData()` 绑定数据：

```
/** index.js */
Page(){
  ...,
  onLoad: function () {
    var that = this;
    wx.request({
      url: CONFIG.API_URL.SHELF_QUERY,
      method: 'GET',
      data: {
        img_size: 'small'
      },
    },
```



```

success: function (res) {
  that.setData({
    shelfNavList: res.data.objects
  });
}
});
},
...
}

```

货架列表就完成了。

四、 template 模板的使用

我们再来看看，首页的分类列表和货架分类列表页两个页面的对比：

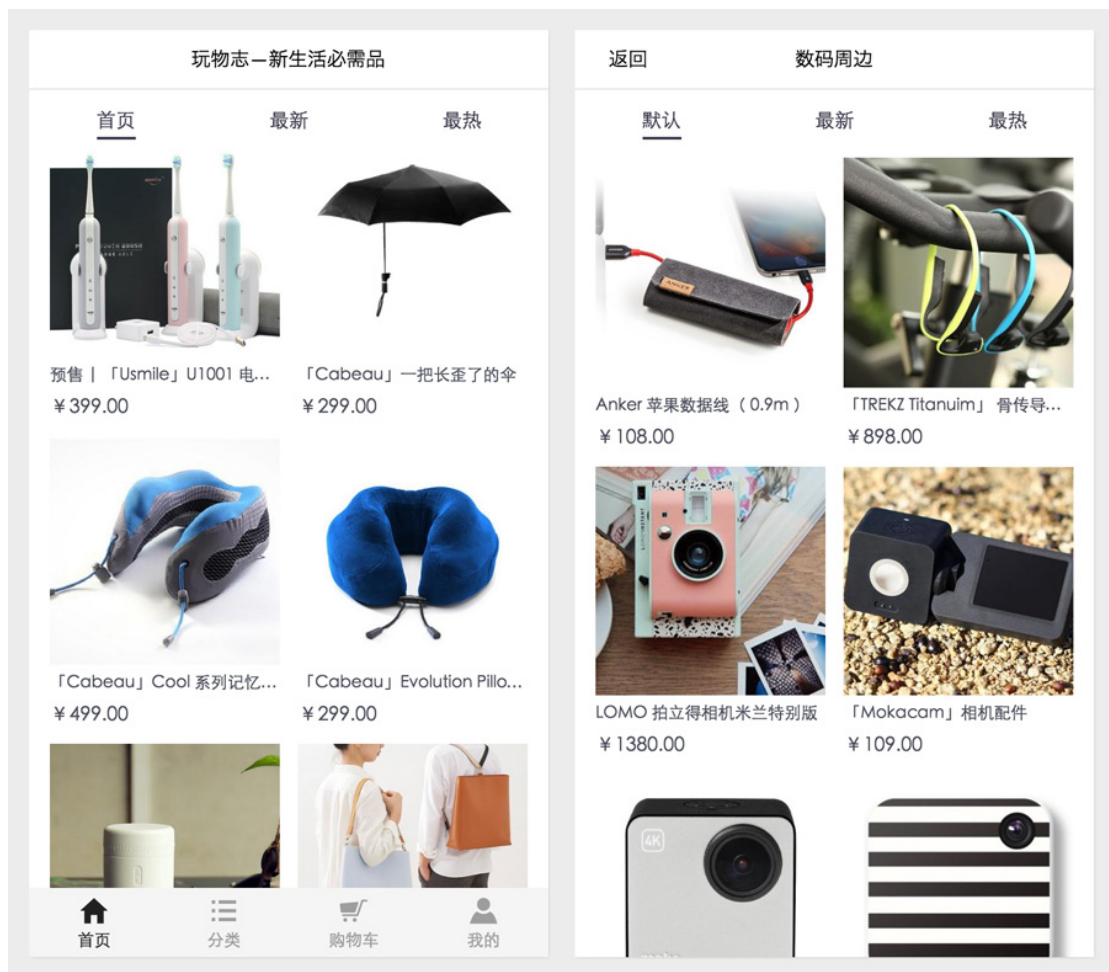


图 附-3-2-3 首页分类列表和货架分类列表对比

不难发现，这两个页面的列表的样式是一模一样的。

那么，我们就可以把每个商品视为一个模块，暂且把这个商品模块称为 `product-card`，我们可以将它用 `template` 封装起来，在需要的地方就将它引入并进行调用。

这里先要弄清楚一个概念，**template** 的主要功能更多的是在于定义一个 **WXML** 代码片段，然后在不同的地方调用。

`template` 拥有自己的作用域，而且只能使用 `data` 传入的数据。

我们来看下 `template` 到底如何使用。

1. 定义 `template` 模板

为了方便代码组织，我们在 `templates` 目录下，新建一个 `productCard` 文件夹，并在 `product-card` 文件夹下新建 `productCard.wxml` 和 `productCard.wxss` 文件。

代码如下：

```
<template name="productCard">
  <view class="product-card">
    <navigator
url="../detail/detail?id={{ id }}">
      <view
style="background-image:
url('{{ cover_image }})" class="product-cover">
</view>
      <view>
        <text
class="product-title">{{ title }}</text>
        <text
class="product-price">
¥{{ price }}</text>
      </view>
    </navigator>
  </view>
```

```
</template>
```

使用 `name` 属性，定义模板的名字，然后将代码片断保存在 `template` 中。

2. 引入 `template` 模板

以首页为例，当要使用到 `productCard` 模板时，我们只需要使用 `import` 引入模板。

在需要显示的位置，外层用 `wx:for` 循环渲染列表，`template` 为子项，使用 `is` 声明需要的使用的模板，用 `data` 传入数据：

```
/** index.wxml */  
  
<import  
src="../../templates/productCard/productCard.wxml  
" />  
<view class="product-list">  
  <block wx:for="{{ productNewList }}" <template  
is="productCard" data="{{ ...item }}" /> </block>  
</view>
```

留意 `data="{{ ...item }}"` 的写法，`item` 是 `wx:for` 中代表数组当前项的默认变量名，`item` 前面的 `...` 操作符相当于 ES6 中的展开运算符，可用于需要解构赋值的地方，没有了解过展开运算符和解构（**Destructuring**）的同学可以先了解一下它们的基本概念。

通过解构，`template` 中就可以直接写成 `{{ id }}`，`{{ cover_image }}`，而不用写 `{{ item.id }}`，`{{ item.cover_image }}`。

它的意义在于实现了 `template` 与 `wx:for-item` 之间的解耦，比如这里设置了 `wx:for-item="product"`，我们只要设改变

`data="{{ ...product }}"` 就可以了。

如果数据没有通过解构，就要将 `template` 的 `{{ item.id }}` 修改成 `{{ product.id }}`，很不方便。

接着，是 `productCard.wxss` 的引入，先在 `productCard` 写好样式，这里就不贴代码了。

五、 模板的 WXSS 文件如何引入

首先请思考一下，在哪里可以引入 `WXSS` 文件？

一种方法是在用到 `productCard` 模板的页面里引入，在这里是在 `list.wxss` 中 `import` 进来。

另一种方法是，直接在 `app.wxss` 中引入。相比较于上一种方法，这个方法只需要一次引入，而所有用到 `productCard` 模板的页面都不用再去引入 `productCard.wxss` 了。

```
/** app.wxss */
@import
"./templates/productCard/productCard.wxss";
```

获取商品列表数据，渲染视图

先通过 `onLoad` 的 `options` 取得货架 `id`:

```
/** list.js */
onLoad: function (options) {
  var id = options.id;

  this.setData({
    shelfId: id
  });
}
```

最后，调用 `wx:request()` 获取商品列表数据，通过 `setData()` 设置 `productNewList`，即可在视图层渲染出完整的

列表。

```
/** list.js */
onLoad: function (options) {
  ...
  wx.request({
    url: CONFIG.API_URL.PRODUCT_LIST,
    method: 'GET',
    data: {
      shelf__id: this.data.shelfId
    },

    success: function (res) {
      if (res.status == 200) {
        that.setData({
          productNewList: res.data.objects
        });
      }
    }
  });
  ...
}
```

到这里，一个像样的列表页面就做好了。但如上面的代码所示，现在小程序会一次性将所有商品列表查询并渲染至页面中。

现实中，这种做法显然是不科学的。我们还需要一个「加载更多」的功能。

这个功能要求我们做到：用户访问时，页面首次加载 20 个商品，点击列表底部的「查看更多」按钮可再多加载 10 个商品。

具体如何实现呢？请继续关注下一期的内容。

本文作者许诺会在「知晓程序 × 开发交流」微信群中和大家交流小程序开发经验。关注知晓程序（微信号 **zxcx0101**），点击底部菜单栏的「加群交流」，即可获得加群方法。

数据加载和图文排版

这里是「电商类小程序实战教程」的第 3 篇文章。

上一期，知晓程序（微信号 `zxcx0101`）以爱范儿旗下的玩物志小程序为 **Demo**，介绍了商品列表的实现方法。

现在，我们将先完成上一期「加载更多列表」的功能，再来谈一谈如何在商品详情页中，将商品描述从富文本数据转换成 **JSON** 数据后，并进行图文排版。

一、 「查看更多」功能

我们还是先来看一下效果图：



图 7-3-1 「加载更多」功能效果图

1. 调用 Toast 提示

通过调用 `wx.showToast()`，可以实现上图中部的 Toast 提示（加载中）。

微信提供的 Toast 组件有 `success` 和 `loading` 两种状态（旧版本的 `loading` 组件已经被废弃）。

```
/**list.js**/  
wx.showToast({  
  title: '成功', //提示的文字信息
```

```
icon: 'success', //图标, 只支持"success"、"loading"  
duration: 2000, //提示的延迟时间(毫秒), 最大为 10000  
success: function() {}, //调用成功的回调函数  
fail: function() {}, //调用失败的回调函数  
//接口调用结束的回调函数 (调用成功、失败都会执行)  
complete: function() {}  
})
```

需要注意的是, 当 `showToast` 执行时, 整个页面是不可点击的。

所以, 不用考虑用户意外触发多次「查看更多」的问题。

调用 `wx.hideToast()` 即可隐藏 `Toast` 提示框。

现在, 我们来思考一个问题。上面说到 `wx.showToast` 的 `duration` 最大值为 10000 毫秒, 也就意味着当请求时间大于 10 秒, 若没有返回数据时, `loading` 提示框会自动消失。

这会带来一些问题:

- 「查看更多」按钮重新变回可点击状态, 可能造成请求的重复提交。
- 加载提示消失与数据成功加载之间出现时间差。

如何解决这个问题呢? 其实很简单。

我们先给 `loading` 设置 10000 毫秒延迟时间, 超过 10000 毫秒后, 即使请求成功, 也不保存任何数据。

同时, 将按钮文字改回「查看更多」即可。

我们初始化一个 `timeoutFlag` 的对象, 使用 `timer =`

`setTimeout()` 设置 9800 毫秒延时。9800 毫秒后，以 `timer` 为 `key` 设置一个对象 `{ isTimeout: true }`，每次发起请求时都会有一个 `timer` 作为标识。

若请求成功时 `timeoutFlag.timer` 的值为 `{ isTimeout: true }`，则不处理请求成功后的数据。

为什么要设置 9800 毫秒而不是 10000 毫秒呢？

那是因为 `setTimeout` 的计时并不准确，我们给出 200 毫秒的提前量，以防止代码出错。

我们来封装一个 `_setTimeoutFlag` 方法：

```
/**list.js**/  
_setTimeoutFlag: function () {  
  var that = this;  
  var timer = null;  
  
  timer = setTimeout(function () {  
    that.setData({  
      tipText: '查看更多'  
    });  
  
    timeoutFlag.timer = { isTimeout: true };  
  }, 9800);  
  
  timeoutFlag.timer = { isTimeout: false };  
  
  return timer;  
}
```

2. 构建与实现 `loadMore` 函数

我们需要一个 `loadMore` 函数，来做到数据的分页查询。

一般，都是通过设置两个查询参数来控制，分别是每次请求条数 `limit` 和每次请求起始数 `offset`（即从第几条获取数据）。

以我们的需求为例：

- 第一次获取 20 条：limit=20 & offset=0
- 第二次再新增 10 条：limit=10 & offset=20
- 第三次再新增 10 条：limit=10 & offset=30

当然，根据各自不同的分页实现，limit 和 offset 的命名也不相同，我们只要了解原理即可。

根据以上逻辑，当用户点击「查看更多」按钮后，小程序在不同情况下应该给出相应的反应：

- 按钮文字变为「正在加载...」，页面弹出「加载中...」的 loading 提示框，设置 offset + 10 并发起请求；
- 有新数据加载时，按钮文字变回「查看更多」，关闭 loading 提示框，保存当前 offset；
- 无新数据加载时，按钮文字变为「没有更多啦」，关闭 loading 提示框。
- 请求失败时，按钮文字变为「查看更多」，关闭 loading 提示框

先在 list.wxml 中设置按钮，并绑定点击事件 loadMore；。

```
/**list.wxml**/  
<view class="btn-load-more" bindtap="loadMore">  
  {{ tipText }}  
</view>
```

根据上面的流程，我们来封装一下 _getProductList 和 loadMore 方法。

```
/**list.js**/  
loadMore: function () {
```

```

var that = this;

this.setData({
  tipText: '正在加载...'
});

wx.showToast({
  title: '加载中...',
  icon: 'loading',
  duration: 10000
});

var timer = that._setTimeoutFlag();

this._getMoreProductList(timer);
},

_getMoreProductList: function (timer) {
  var that = this;

  wx.request({
    url:          CONFIG.API_URL.PRODUCT_LIST          +
this.data.orderBy,
    method: 'GET',
    data: {
      limit: 10,
      offset: this.data.offset + 10,
      shelf__id: this.data.shelfId,
      img_size: 'small'
    },

    success: function (data) {
      if (!timeoutFlag.timer.isTimeout) {
        if (data.data.objects.length > 1) {
          var objects = that.data.productList
            .concat(data.data.objects);

          that.setData({
            productList: objects,
            offset: that.data.offset + 10,
            tipText: '查看更多'
          });
        }
      }
    }
  });
}

```

```
    });  
  } else {  
    that.setData({  
      tipText: '没有更多啦'  
    });  
  }  
}  
clearTimeout(timer);  
wx.hideToast();  
},  
  
fail: function () {  
  that.setData({  
    tipText: '查看更多'  
  });  
  wx.hideToast();  
}  
});  
}
```

至此，我们已经完整实现了商品列表页的开发，接下来进入商品详情页。

二、商品详情页与图文排版实现

在商品详情页中，我们主要来说一说图文排版的实现。

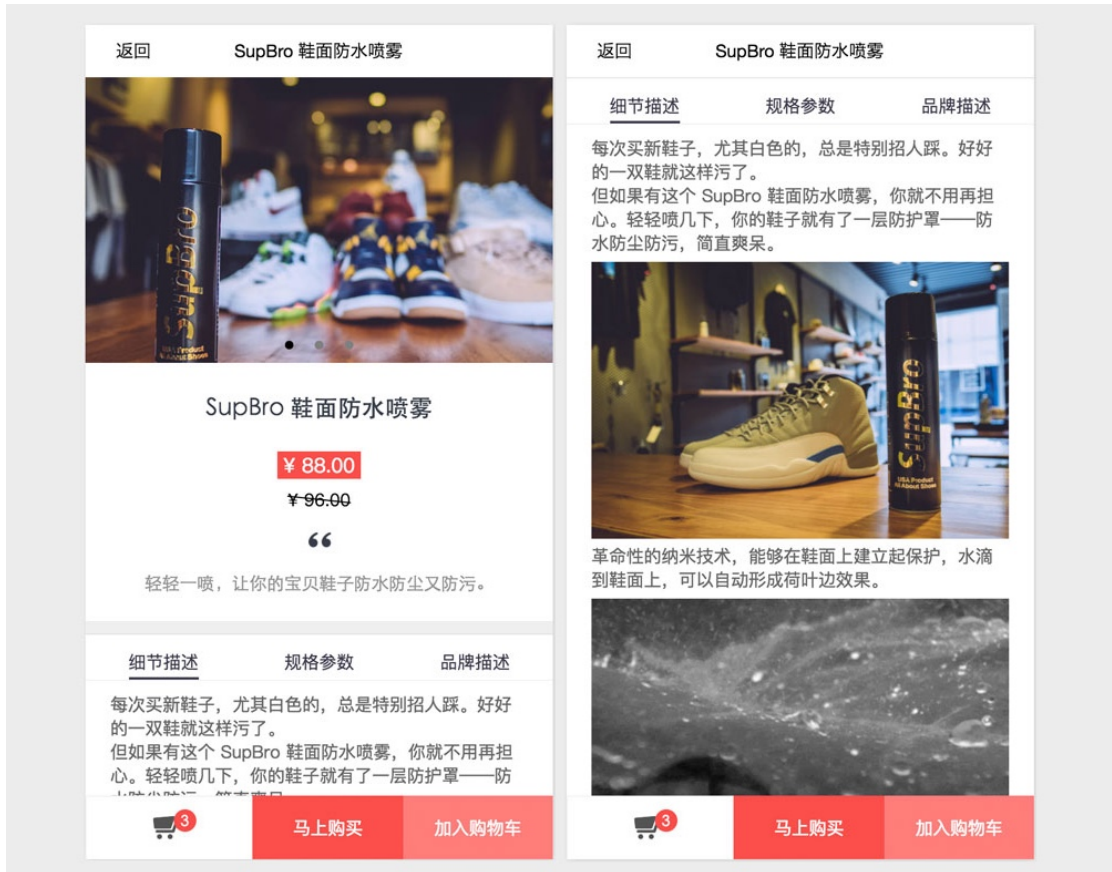


图 7-3-2 图文排版最终效果

目前商品详情的图文介绍, 一般都是通过后台的富文本编辑器来排版创建的, 数据都是以 HTML 代码的形式保存下来, 然后在页面上直接渲染出来。

但是对于小程序来说, HTML 形式的数据不能直接拿来使用, 因为小程序根本不能解析 HTML 代码。

此时, 我们需要处理由富文本编辑器所创建的数据, 将其转换为 JSON 格式的数据, 以便于在小程序里使用。

我们有两种思路去解决这个问题:

由前端去解析 HTML 格式的数据, 用正则匹配 HTML 标签, 然后转换成 JSON 数据;

交由后端去解析，同时新增一个专用的 API，返回解析过的 JSON 数据。

我们选用的是第二种方案，获取到的 JSON 数据格式如下：

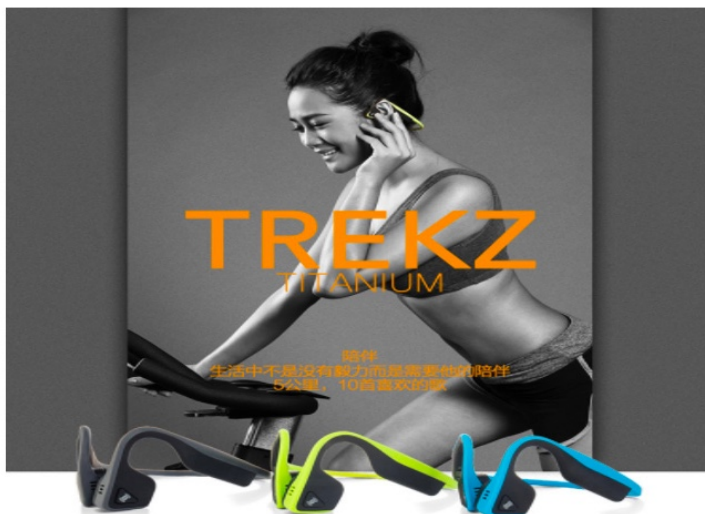
```
{
  "description": [
    {
      "content": "xxx.jpg",
      "type": "image"
    },
    {
      "content": "商品描述文字",
      "type": "text"
    }
  ]
}
```

我们只要将获取的数据用 `wx:for` 渲染出来，渲染时判断 `type` 选择对应的 `text` 或 `image` 组件即可。

```
/**detail.wxml**/  
<block wx:for="{{ description }}">  
  <view  
    class="description-img"  
    wx:if="{{ item.type == 'image' }}">  
    <image  
      class="description-img-content"  
      src="{{ item.content }}">  
    </image>  
  </view>  
  <view  
    class="description-text"  
    wx:elif="{{ item.type == 'text' }}">  
    <text class="description-text-content">  
      {{ item.content }}  
    </text>  
  </view>  
</block>
```

来看效果图：

返回 「TREKZ Titanuim」 骨传导运动耳机



页岩灰



常青藤绿



深海蓝

10



马上购买

加入购物车

图 7-3-3 图文排版功能实现后的效果

仔细观察一下，我们可以发现，每一张图片的宽高都发生了不同程度的裁剪、拉伸或压缩。

这是由于 image 组件的默认宽高导致的，我们在第一期说过，要

特别留意小程序内置组件的默认样式，它们很可能会引起各种各样的问题。

我们的期望是，图片能够按照原有的宽高比例进行显示，并且每一张图片的宽度都能撑满整个内容区域。这应该如何处理呢？

这里就需要用到 `image` 组件的 `load` 事件，使用 `bindload` 属性将事件绑定在 `image` 组件中。

我们可以用 `event.detail = {height:' 图片高度 px', width:' 图片宽度 px'}` 的方式调整图片宽高尺寸。

我们定义一个 `imageLoad` 函数，作为 `image` 组件的 `load` 事件的处理函数，用 `data-index` 保存当前 `wx:for` 的 `index` 值作当前图片的索引值。

`detail.wxml` 改写 `image` 组件如下：

```
<image
  class="description-img-content"
  src="{{ item.content }}"
  bindload="imageLoad"
  data-index="{{ index }}">
</image>
```

`detail.js` 定义 `imageLoad` 函数如下：

```
data: {
  images: {}
},
imageLoad: function (e) {
  var realityWidth = e.detail.width;    //获取图片实际宽度

  var realityHeight = e.detail.height;  //获取图片实际高度
```



```

//获取图片的实际宽高比例
var proportion = realityWidth / realityHeight;

//设置图片的渲染宽度, 750rpx - (2 * 40rpx) = 670rpx;
//父元素的左右 padding 都是 40rpx
var viewWidth = 670;

//按比例计算出图片的渲染高度
var viewHeight = 670 / proportion;
var images = this.data.images;

//将图片的 data-index 值作为 images 对象的 key
//用来储存该图片的计算后的渲染宽高
images[e.currentTarget.dataset.index] = {
  width: viewWidth,
  height: viewHeight
};

this.setData({
  images: images
});
}

```

此时，我们就可以在 image 组件中获取到 `images[index].width` 和 `images[index].height`，并通过设置行内的 `style` 给每一个图片设置宽高了。

detail.wxml 改写 image 组件如下：

```

<image class="description-img-content"
src="{{ item.content }}" bindload="imageLoad"
data-index="{{ index }}"
style="width:{{ images[index].width }}rpx;
height:{{ images[index].height }}rpx">
</image>

```

完成的效果图如下：

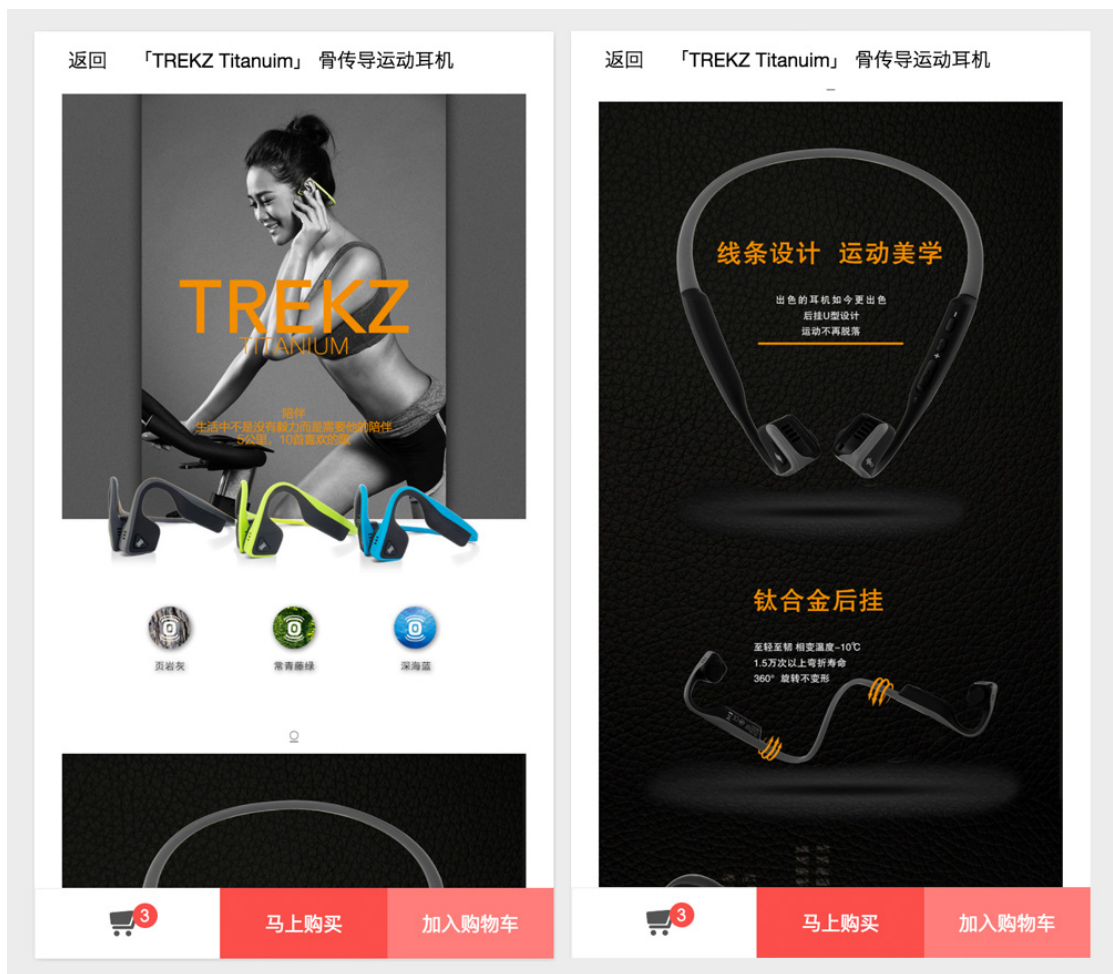


图 7-3-4 最终效果图

三、 本期内容总结

本期的「电商类小程序实战教程」就到这里了。

想跟本文作者在线交流小程序开发，关注微信号 **zxcx0101**，点击下方菜单栏「加群交流」即可。

最后，知晓程序(微信号 **zxcx0101**)来总结一下本期内容的重点：

1. Toast 提示：

- Toast 提示可以使用 `wx.showToast()` 来调用，使用 `wx.hideToast()` 隐藏。

- 开发者可以自定义 **Toast** 展示时间、展示文案等。
- **Toast** 最长展示时间不能超过 **10000** 毫秒，开发者需要做好超时处理工作。

2. 图文排版：

- 如果服务器中的数据是 **HTML** 数据，需要转换为 **JSON**，小程序才能解析。
- 直接将图片进行排版，图片宽高会出现问题。开发者需要处理好图片的宽高问题。

在下一期中，我们会结合 **navTab** 和 **tabBar** 组件，来谈一谈自定义组件的实现方式，请大家继续关注知晓程序。

写在最后

《微信小程序入门指南》已经告一段落了，但这本书还会继续更新。我们会在这里，给大家持续分享小程序的最新资讯和开发指南。

无论你是一个开发者，还是一个普通用户，我们都坚信：小程序必将对你的生活造成巨大的影响。

It is a time of ambitious plans and endless possibilities.

你可以关注我们的公众号（微信号：[zxcx0101](#)），及时了解小程序的最新动态；也欢迎你加入知晓程序社区（[minapp.com](#)），跟小程序同道人一起交流讨论。

不论你是小程序开发者，或是希望了解小程序的企业主，通过邮箱 bigbang@ifanr.com 就能跟我们取得联系。

附录

小程序大事记

从「应用号」到微信「小程序」，微信官方在小程序方面都做了哪些动作？

知晓程序（微信号 **zxcx0101**）将所有的官方资讯，整理成了这份小程序大事记，让你一次性了解微信小程序的前世今生。

小程序时间线

- 2016 年 1 月 11 日：应用号初现、开发工具发布
- 2016 年 1 月 20 日：WeUI 发布
- 2016 年 4 月 19 日：微信内置浏览器升级
- 2016 年 9 月 22 日：小程序发布内测
- 2016 年 10 月 27 日：小程序新增 19 接口
- 2016 年 11 月 3 日：小程序开放公测
- 2016 年 11 月 18 日：小程序入口谜团
- 2016 年 11 月 22 日：开发者工具更新

2016 年 1 月 11 日：应用号初现、开发工具发布

张小龙现身「微信公开课 Pro 版」，首次发表公开演讲。也是在此次演讲中，微信官方正式宣布正在开发「应用号」。



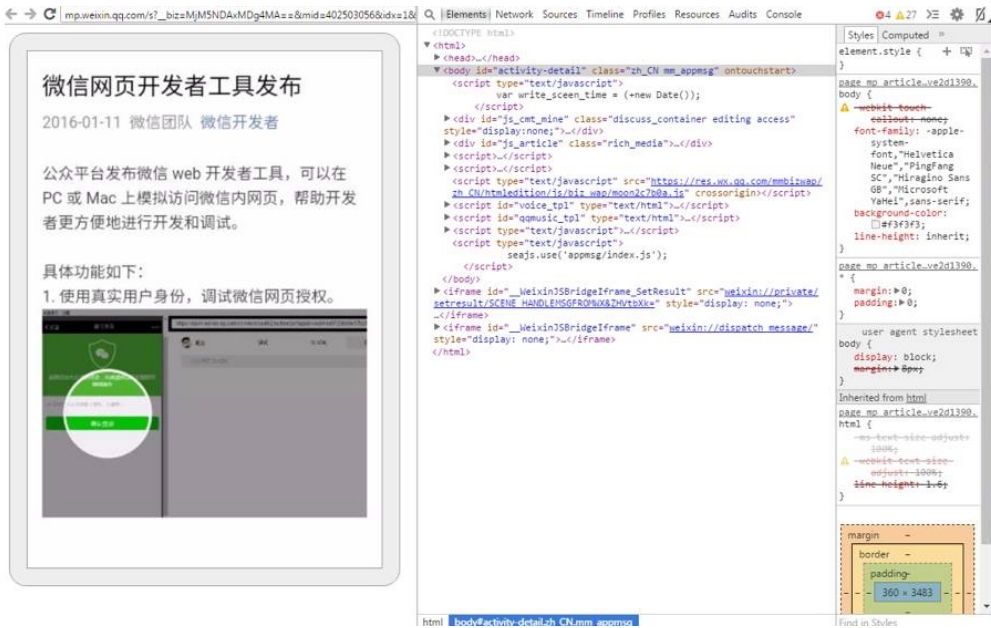
图 附-1-1 张小龙在「微信公开课 Pro」演讲

张小龙称，这是微信的「新的形态」。

我们希望存在一种新的公众号形态，这种形态下面用户关注了一个公众号，就像安装了一个 app 一样。他要找这个公众号的时候就像找一个 app，在平时这个号不会向用户发东西的，所以 app 就会很安静的存在那里，等用户需要的时候找到它就好了。

张小龙此次「剧透」引来了互联网行业多方猜测，应用号也被形容为「像 app 一样的公众号」。大家纷纷开始猜测应用号会如何出现在微信中。

同日，微信宣布发布「微信 Web 开发者工具」，该工具可以让开发者在电脑上模拟访问微信内网页，帮助他们更方便地进行开发和调试。该工具后来成为应用号（小程序）的开发集成环境（IDE）。



2016 年 1 月 20 日：WeUI 发布

微信发布自己的网页设计样式库 WeUI，开发者可以使用它快速开发出符合微信 UI 界面标准的网页。

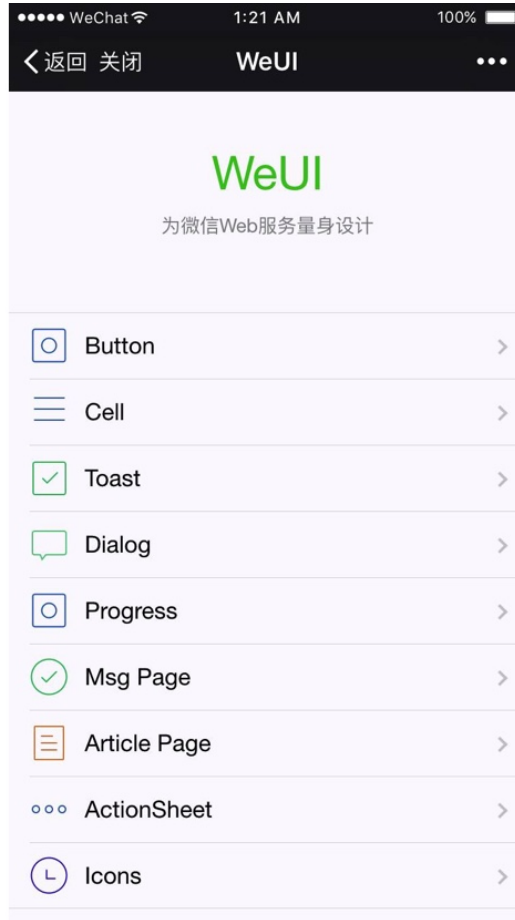


图 附-1-3 WeUI 演示页面

最初版本的 WeUI 包含了包含了移动应用中所能使用到的所有元素，包括按钮、表单、Toast 提示、对话框、进度条等多种预置样式。

WeUI 控件样式库奠定了小程序设计规范的基础。

2016 年 4 月 19 日：微信内置浏览器升级

微信更新「微信 Web 开发者工具」至 0.5.0 版本，该版本的开发者工具支持远程调试。同时，微信宣布 Android 版微信内置浏览器全量升级为 X5 Blink 内核。

微信官方宣称，这种内核可以让微信内浏览器具有更好的 HTML 5 / CSS 3 支持、强大的渲染能力，同时提供了硬件状态检测功能。

2016 年 9 月 22 日：小程序发布内测

微信悄然将「应用号」更名为「小程序」，并陆续发放内测邀请。

微信在官方邀请函中写到：

我们提供了一种新的开放能力，开发者可以快速地开发一个小程序。小程序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。



图 附-1-4 微信小程序邀请函

在首批开放的 API 接口中，小程序具有基本的视觉组件、页面导航、画布渲染、网络访问、文件处理、数据缓存、位置数据、设备信息、界面控制、用户体系等能力。

同日，张小龙在其微信朋友圈发布「什么是小程序」的短文。

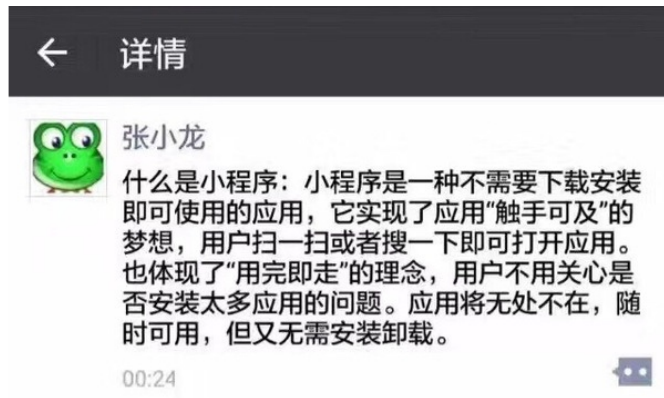


图 附-1-5 张小龙在朋友圈阐述什么是小程序

2016 年 10 月 27 日：小程序新增 19 接口

微信小程序官方 IDE「微信开发者工具」更新，宣布为小程序新增 19 种 API 接口。

新增的 19 中接口中，包括增强的交互反馈功能、数据与文件管理、地理位置选择、获取图片信息、绘图、音频媒体、页面栈处理等逻辑接口，以及包括 `<textarea />` 在内的多种视觉元素组件等。

2016 年 11 月 3 日：小程序开放公测

微信小程序宣布进入公测状态。公测期间，所有企业、政府、媒体和其他机构都可以登记注册小程序。

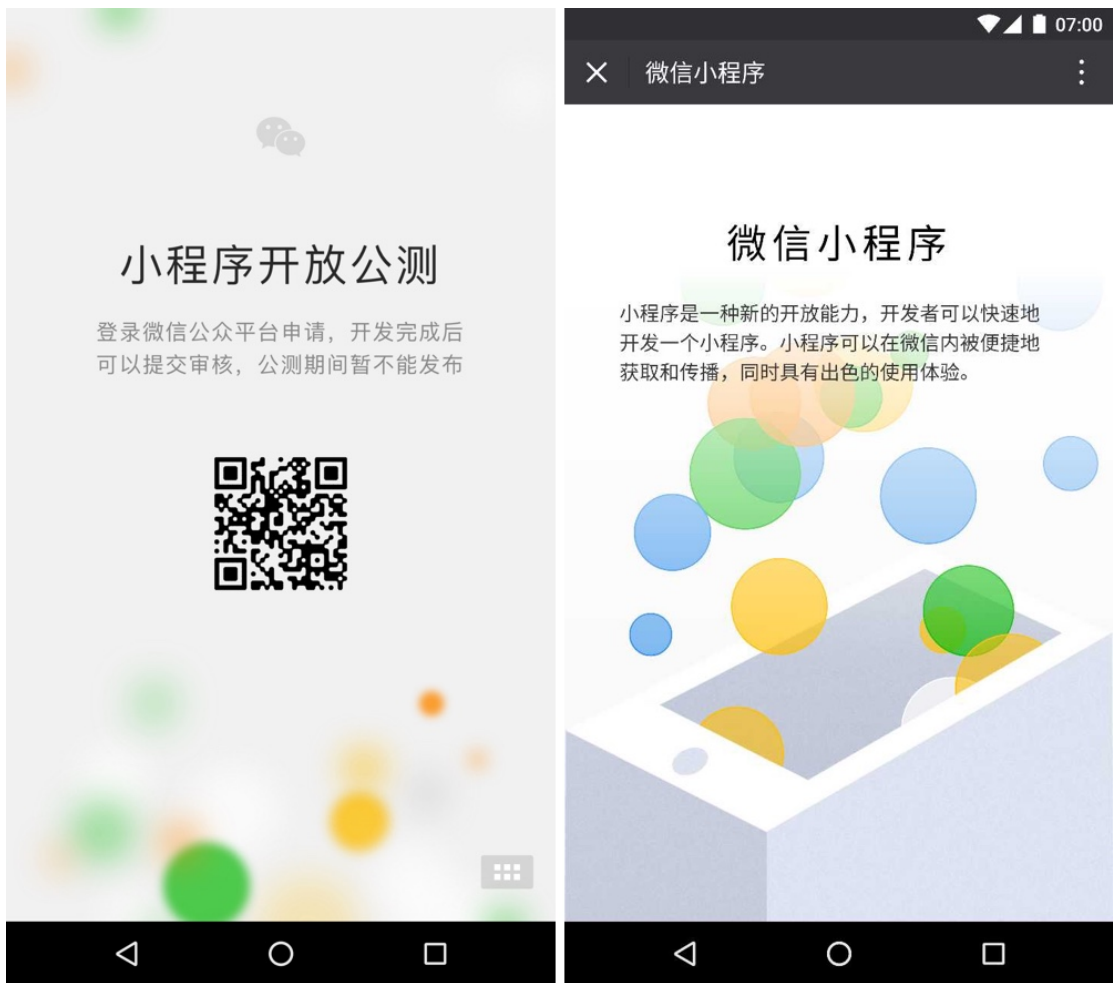


图 附-1-6 微信小程序公测宣传

公测期间，开发者可以将小程序提交审核，但审核通过的小程序暂时不能公开使用。

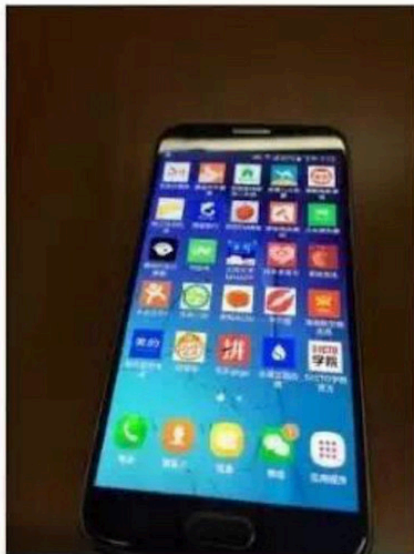
2016 年 11 月 18 日：小程序入口谜团

张小龙发布朋友圈照片，照片中有一台手机。这台手机的屏幕显示着一个摆满应用图标的桌面。



Allen Zhang

程序猿的一小步，程序的一大步



27分钟前



图 附-1-7 张小龙朋友圈展示安装了小程序的手机桌面

从屏幕上的应用名称，到图片附文「程序猿的一小步，程序的一大步」来看，微信小程序在 **Android** 设备上极有可能以原生应用的形态存在。

2016 年 11 月 22 日：开发者工具更新

微信小程序官方 IDE「微信开发者工具」更新，增强了网络调试的体验。

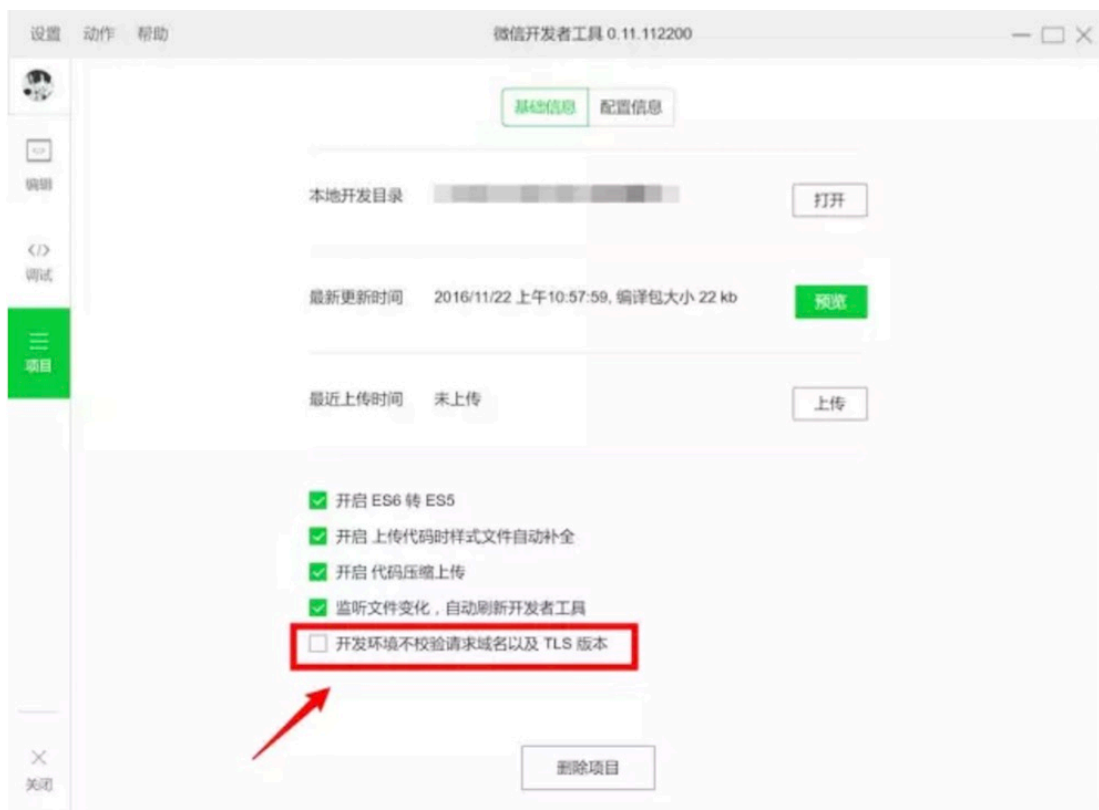


图 附-1-8 微信开发者工具不再限制开发环境的请求域名

在新版本的开发者工具中，后端开发者可以在开发期间，专注于开发工作本身，无需考虑环境搭建问题，进一步增强了网络调试的体验。同时，开发者也可以在工具中查看到白名单域名，提高开发效率。

在小程序正式发布之前，还会有什么大动作呢？我们拭目以待。