

微信小程序开发工具安装指南及注意事项

这两天闲来无事，也安装了“微信折叠”的开发工具来玩一下。以下是一些小道消息及使用体验，过两天我会写一篇文章以开发者的角度来详细评价微信小程序：

- 微信小程序不能开发游戏类、直播类功能，小程序每个人关注的上限是 20 个（还不确定，不过我相信这是真的，这次公布的 API 里并没有视频组件。微信太大，苹果要有所顾忌，但是微信也要做出相应的让步）
- 微信目前有没有同苹果商谈好，还是个未知数，毕竟会对 AppStore 有一定的冲击。
- 抛弃了大量的 javascript 组件后，这个生态体系变得相当的封闭，微信解释肯定是：为了更好的性能提升。那么我们拭目以待。
- 小程序的入口是微信里的三级菜单，就是在“Tab 栏发现里的游戏下面加入一个“小程序”。反正，这一栏里的购物和游戏我是从来没点进去过的。
- 以腾讯的尿性，小程序同服务号一样，其关系链及重要功能的开放程度会因“人”而异。对，优质的接口只会开放给腾讯的儿子们（滴滴呀、京东呀）
- 微信从来就是一眼不合就封杀，优步就是很好的案例。官方解释是优步涉嫌诱导，滴滴同样被禁过。利益面前还讲道理？反正我是不信的。腾讯的投资覆盖各个行业，一旦腾讯掌握了应用分发入口，给儿子们一些小小的甜头，应该不是什么大事儿吧。有人说 AppStore 也有很多限制，但至少苹果并没有像腾讯这样大量的投资消费级 App，直接产生利益冲突的点并不多。话说回来，这点创业者是不用考虑的，暂时，你的产品和创意，在微信眼里还是个蝼蚁。。。你不用担心受到不公正的对待。如果真有一天你被微信封杀了，只要不是因为违规内容，那么你应该高兴，然后迅速炒作一波.....
- 很多做 Web 的朋友尝试在小程序里写 div、ul 等标签，别试了，没效果，包括 JQUERY 在内的 DOM 操作库，小程序内是不能运行的。小程序本质是 JS+ReactNative，对 RN 底层做了重构，已经和 dom 说拜拜了。

下面是详细的安装步骤。[win10 下测试通过](#)

以下是安装步骤：

1. 按操作系统下载相应开发工具版本

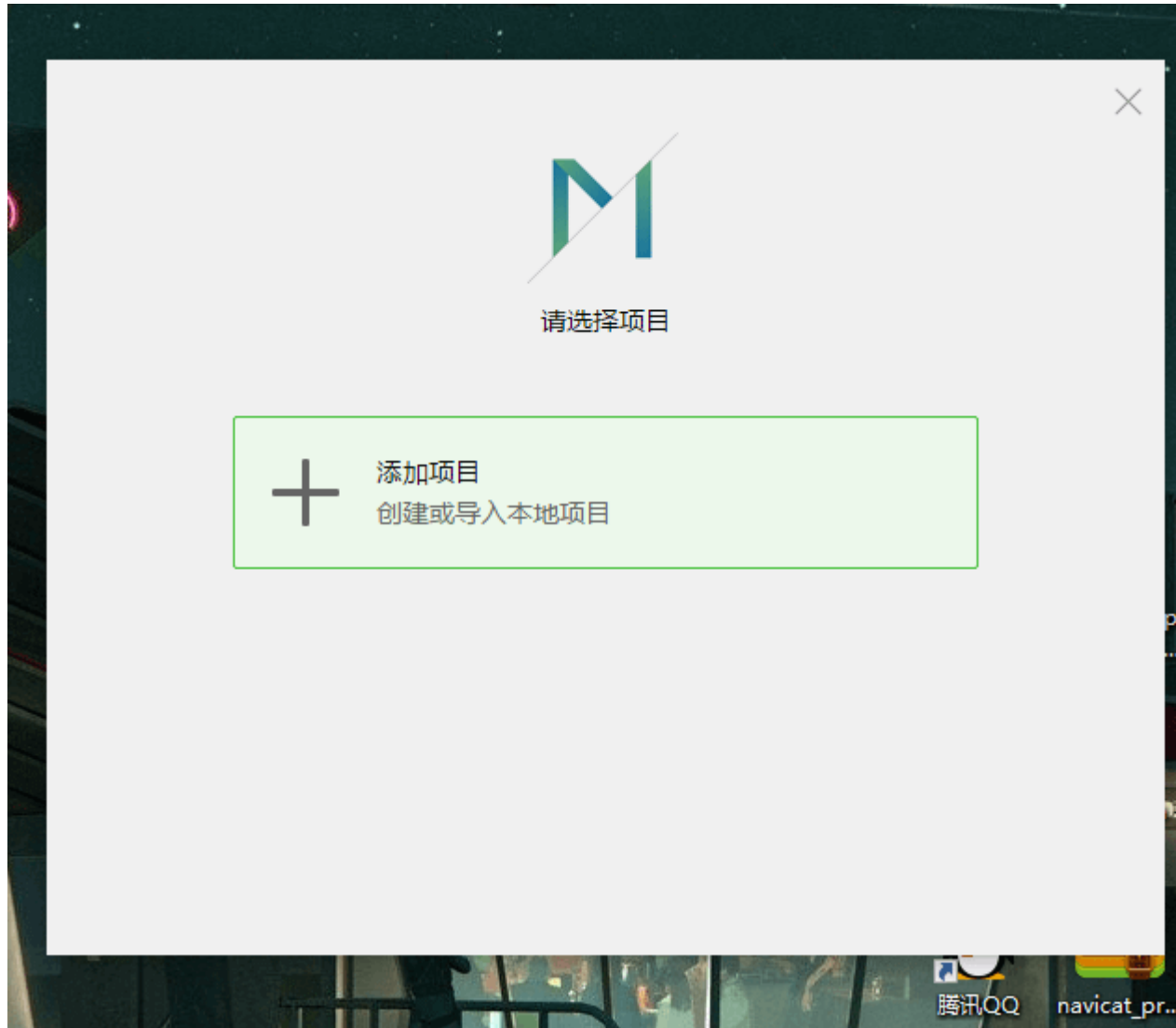
[windows 64 位](#)

[windows 32 位](#)

[mac](#)

[小程序示例组件 demo 下载](#)

2. 双击安装
3. 打开微信 web 开发者工具
4. 添加一个项目，如图



5. 填写项目信息 注意 AppId 这里选择“无 AppId”,AppID 只有内测用户才有，无 AppId 在目前有功能限制，比如不能发布项目到手机微信中预览。

[返回](#)

添加项目

AppID

无 AppID 部分功能受限

返回填写小程序AppID

项目名称

imayi_test

项目目录

D:\Project\App_WX

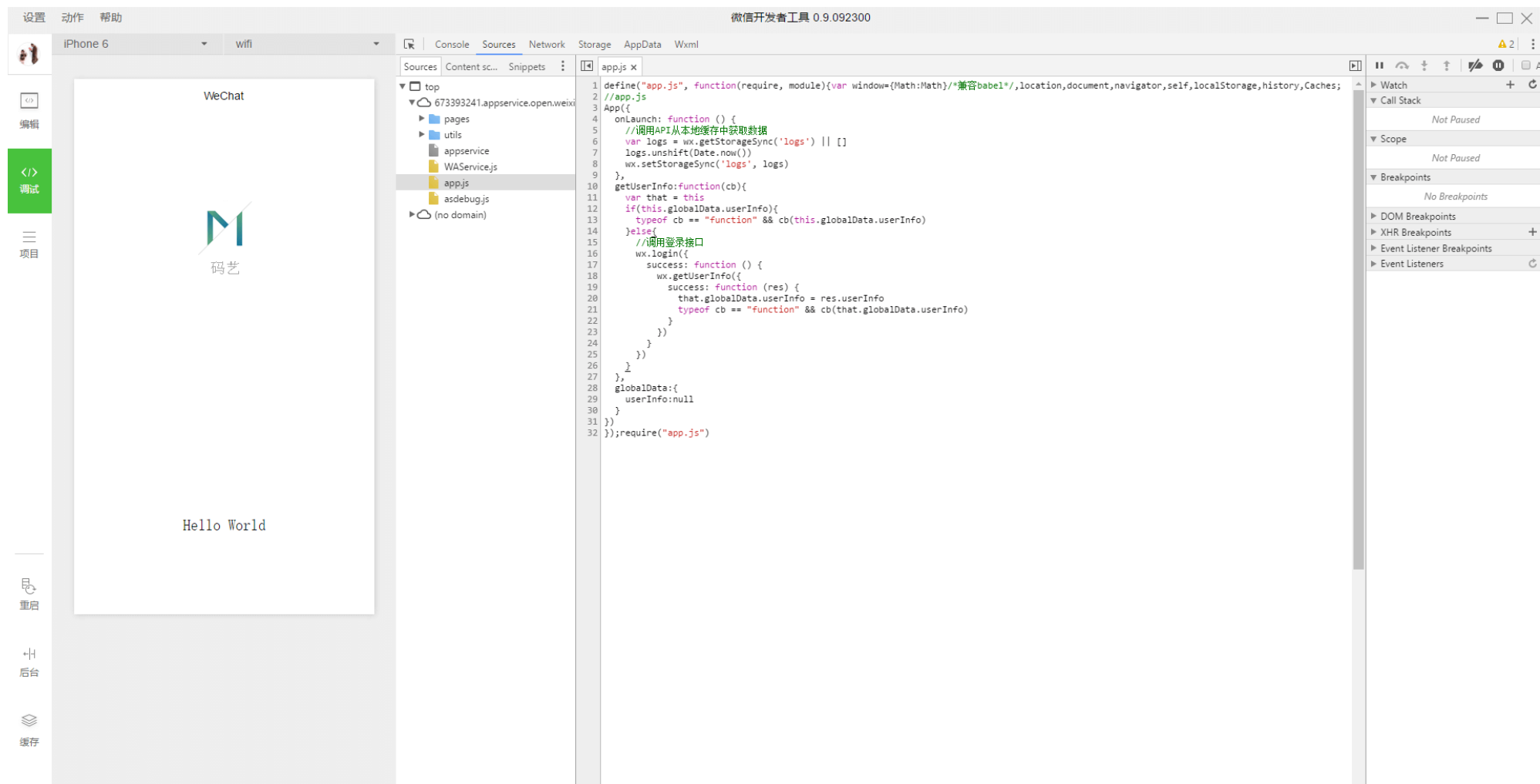
选择

☒ 在当前目录中创建 quick start 项目

取消

添加项目

6. 点击“添加项目”按钮后，将创建一个小程序项目，工具已经为你创建了一个 demo —— Hello World



开发者工具功能区域介绍

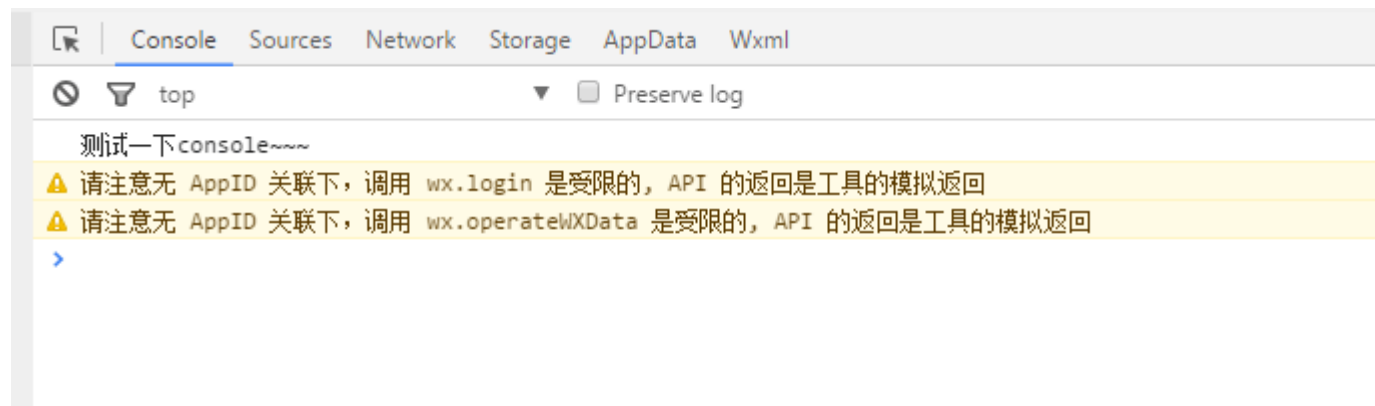
工具左侧分别有编辑、调试、项目调试区按钮，分别解释 3 个区域的作用。

编辑： 是你写代码的地方，左侧的文件目录是微信为你创建的一个项目基础骨架，你可以在这个骨架上编写自己的业务代码。当我们编写完代码后，点击编辑视图左下角的“**编译**”即可进行调试和运行（当点击“编译后”，工具视图会自动跳转到第二个“**调试**”视图）

调试： 是你调试代码的地方。整个界面同 **Chrome** 的调试工具一模一样（应该是内嵌了 **WebKit** 内核），相信有 **Web** 开发经验的同学很快就能上手。**F8** 运行到下一个断点、**F10** 单步运行、**Ctrl+/ **注释代码。****

解释一下“调试”视图下，顶部的六个 Tab 栏（Console、Sources、Network、Storage、AppData、Wxml）：

Console: 工具的输出区域，在程序中调用 `console.log('测试一下 console')`，可在这里输出自定义调试信息。同时，这里也是工具遇到异常和发出警告的信息输出区域。见图：



Sources: 源代码显示区域，这里你可以打断点并调试代码。**Sources Pannel** 用于显示当前项目的脚本文件，同浏览器开发不同，微信小程序框架会对脚本文件进行编译的工作，所以在 **Sources Pannel** 中开发者看到的文件是经过处理之后的脚本文件，同编辑区域的代码有稍许的差异，比如开发者的代码都会被包裹在 **define** 函数中，并且对于 **Page** 代码，在尾部会有 **require** 的主动调用。

WxmlConsoleSourcesNetworkStorageAppData

SourcesContent ... Snippets

app.jsapp.js [sm] x

top

239393028.appservice.open.wei

pages

utils

appservice

WAService.js

app.js

app.js [sm]

asdebug.js

(no domain)

1 //app.js

2 App({

3 onLaunch: function () {

4 //调用API从本地缓存中获取数据

5 var logs = wx.getStorageSync('logs') || []

6 logs.unshift(Date.now())

7 wx.setStorageSync('logs', logs)

8 },

9 getUserInfo:function(cb){

10 var that = this

11 if(this.globalData.userInfo){

12 typeof cb == "function" && cb(this.globalData.userInfo)

13 }else{

14 //调用登录接口

15 wx.login({

16 success: function () {

17 wx.getUserInfo({

18 success: function (res) { res = Object {errMsg: "getUserInfo:ok", rawData: '{"nick

19 that.globalData.userInfo = res.userInfo

20 typeof cb == "function" && cb(that.globalData.userInfo)

21 }

22 }

23 }

24 })

25 }

26 },

27 globalData:{

28 userInfo:null

29 }

30 })

Watch

Call Stack

success app.js [sm]:19

(anonymous function) WAService.js:1

i asdebug.js:1

i asdebug.js:1

d asdebug.js:1

(anonymous function) asdebug.js:1

Paused on a JavaScript breakpoint.

Scope

Local

res: Object

this: Object

Closure (getUserInfo)

Global Window

Breakpoints

app.js [sm]:19

that.globalData.userInfo = ...

DOM Breakpoints

XHR Breakpoints +

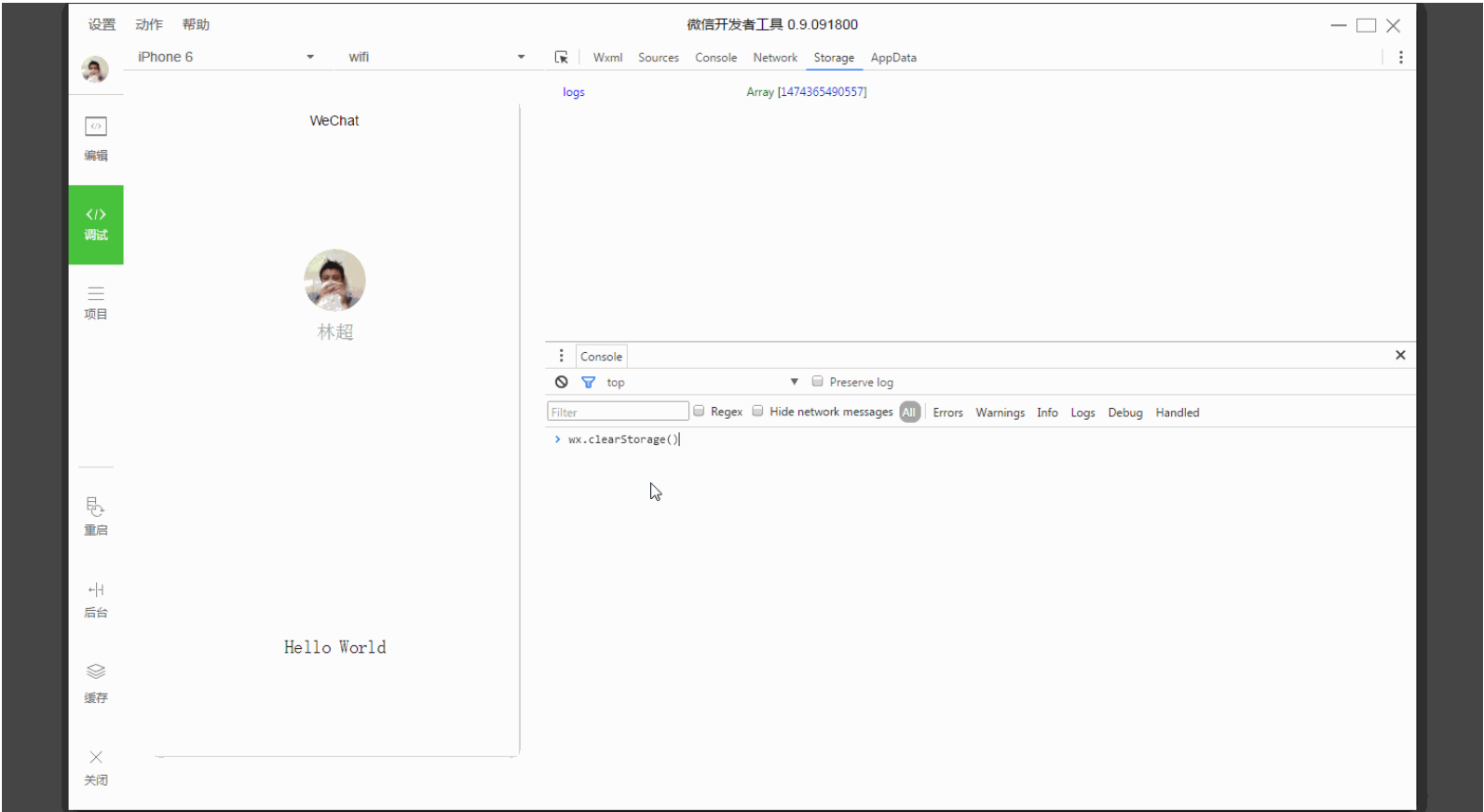
Event Listener Breakpoints

Event Listeners

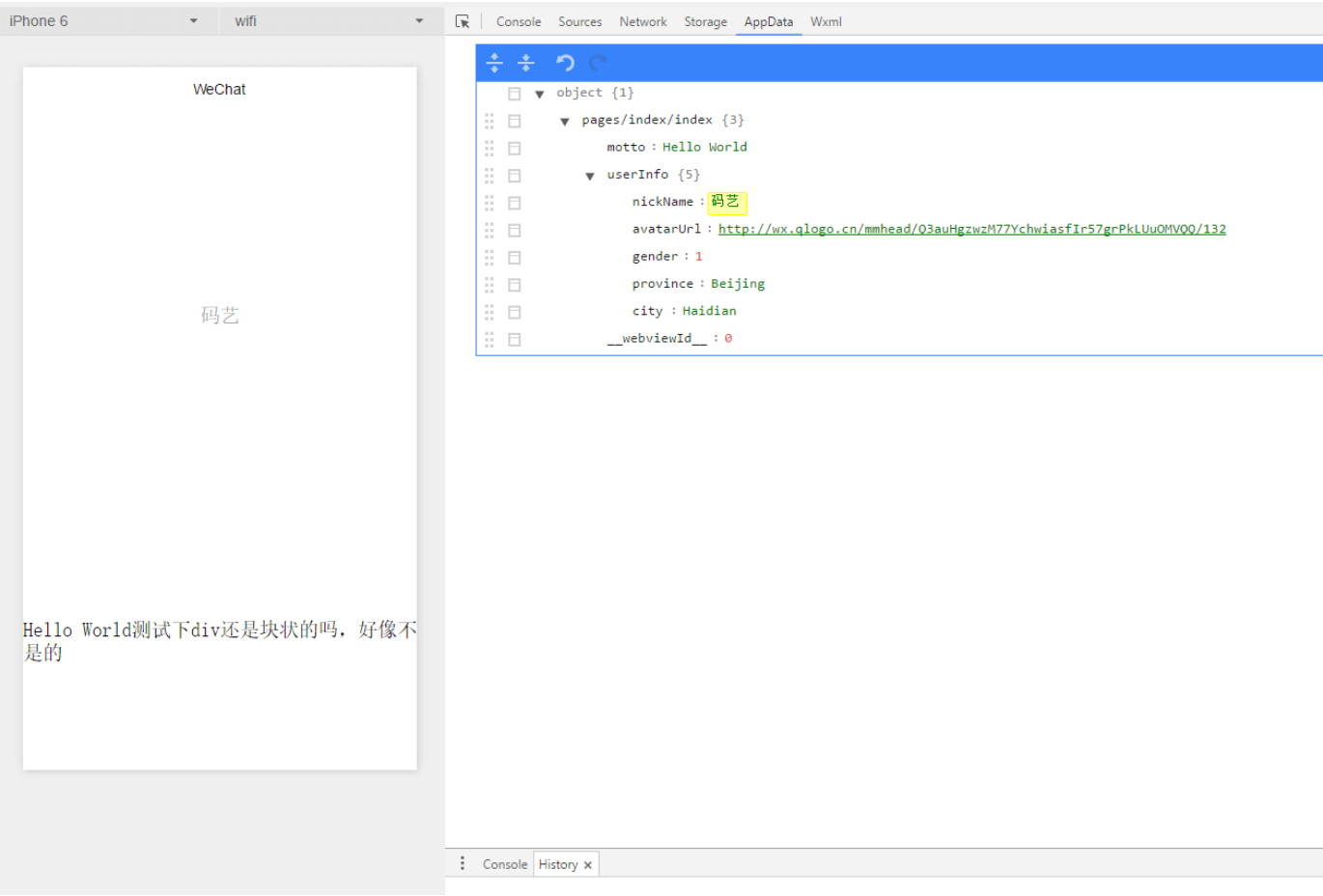
The screenshot shows the Chrome DevTools Network tab. The top toolbar includes icons for Wxml, Console, Sources, Network, Storage, and AppData. The Network tab is active, displaying a list of requests. The first request, 'SendMsg?mmm=123:123&fo...', is selected. The 'Headers' panel is expanded, showing the following details:

- General:**
 - Request URL: `https://stream.weixin.qq.com/weapp/SendMsg?mmm=123:123&foo=bar&mid=123%3A456`
 - Request Method: GET
 - Status Code: 200 OK
 - Remote Address: 183.61.49.165:443
- Response Headers:**
 - Connection: keep-alive
 - Content-Length: 597
 - Content-Type: text/html; charset=UTF-8
 - Date: Mon, 10 Oct 2016 09:48:53 GMT
 - Etag: "da60dd46b00f802119283b658302de6aa557b1b6"
 - Server: nginx
- Request Headers:**
 - Accept: */*
 - Accept-Encoding: gzip, deflate, sdch, br
 - Connection: keep-alive
 - content-type: application/json
 - Cookie: testcookie
 - Host: stream.weixin.qq.com
 - Referer: https://servicewechat.com/touristappid/devtools/page-frame.html
 - User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36 apservice webview/100000
 - X-Requested-With: XMLHttpRequest
- Query String Parameters:**
 - mmm: 123:123
 - foo: bar
 - mid: 123:456

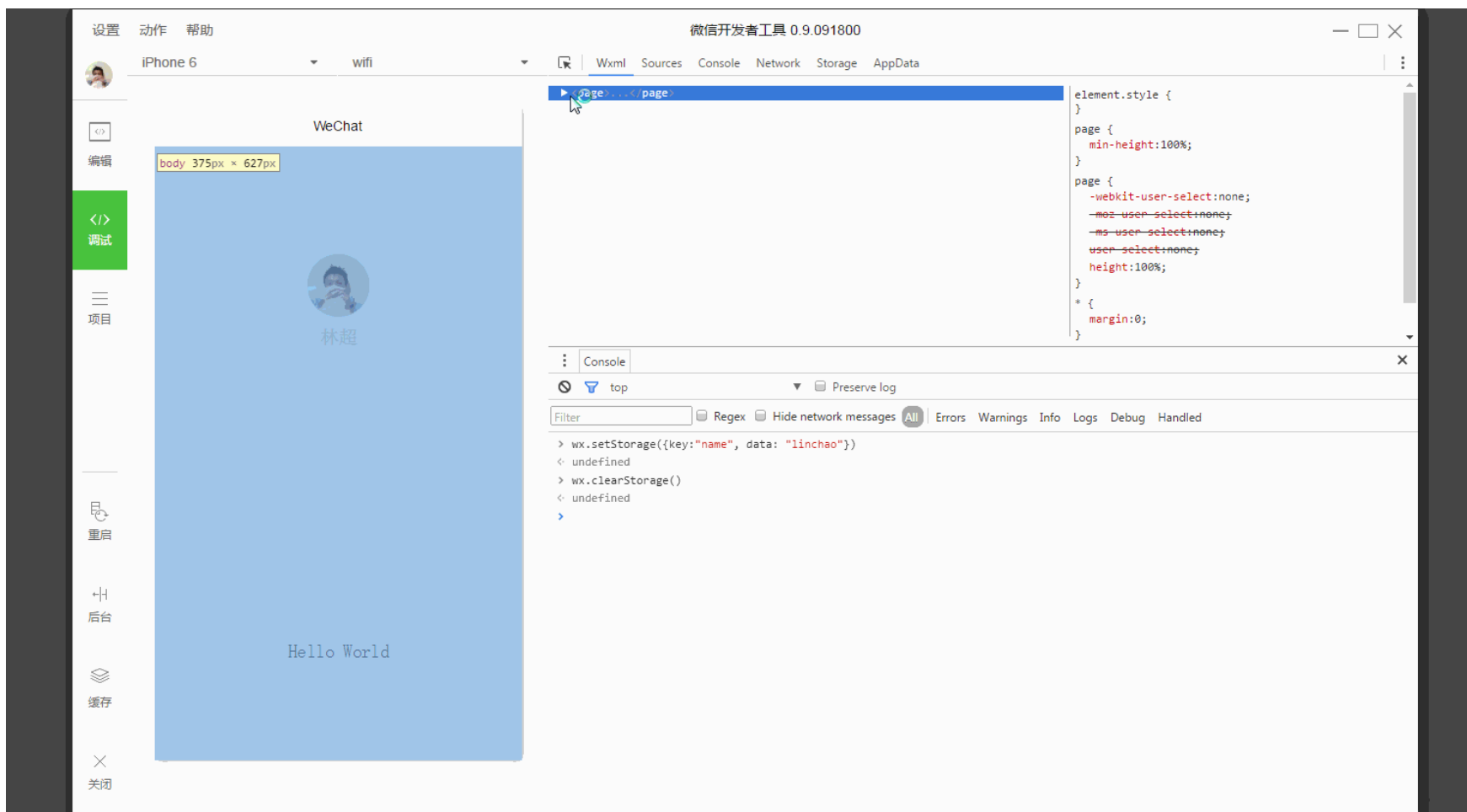
Storage: 如果你在程序中使用了 `wx.setStorage` 或者 `wx.setStorageSync` 后，在这里将会显示你的数据存储情况。是的，小程序支持本地缓存数据。



AppData: 用于显示当前项目当前时刻 `appdata` 具体数据。可以在这里管理应用程序中的各类变量（不同于 `Storage` 中的缓存变量，只有调用 `wx.setStorage` 或者 `wx.setStorageSync` 的数据才会出现在 `Storage` 中）。在这，你可以随时修改变量，工具将把变量改变在左侧预览视图中实时显示。我们可以看到，`AppData` 以页面（`pages/index/index`）作为分组单位，每个页面只会显示自己的 `AppData`，非常的清晰。



Wxml: 将 `Wxml` 标签同预览界面实时关联，在这里可以看到真实的页面结构以及结构对应的 `wxss` 属性，同时可以通过修改对应 `wxss` 属性，在模拟器中实时看到修改的情况。通过调试模块左上角的选择器，还可以快速找到页面中组件对应的 `wxml` 代码。



最后说下“项目”这个功能区块儿，项目实际上是用来上传和预览项目的。微信要求所有发布的小程序都必须审核，所以需要在这里将小程序上传（应该是类似于 AppStore，但肯定不是 AppStore 的发布模式，苹果应该不会允许。小程序目前同用户的接触方式还是未知数，而这恰恰是最重要的一块儿）。如果你没有被内测邀请，这里你是不能上传和预览项目的，也不可以真机运行小程序，所以真正的体验相对于原生 App 还是未知数。目前阶段，你只能熟悉一下小程序的 API，以便在公测后迅速拿出自己的产品，抢夺第一波红利。

总体来说，小程序的大概开发模式和模型 已经暴露，其本质是 JS+RN，很多人说 JavaScript 程序员的春天来了，其实小程序的体系和 Web 开发还是有很多区别的。微信言下之意是要抛弃你们熟悉的 DOM 采用他的组件体系，更熟悉这种组件式开发的不是传统的 Web 前端，而是 iOS 开发人员（更早点你可以追溯到 Flex 和 SilverLight），我本

人是不喜欢这种非 **JS** 原生的组件式开发的，自由度会受限。后续，我会持续更新小程序的开发例子和教程，尽快帮助大家上手开发业务和产品。你现在需要做的，就是静下心来，我应该开发一个什么样的产品？