

微信小程序把玩《一》：window 配置，tabBar 底部导航，应用生命周期 ...

本篇默认已经成功安装微信小程序工具

— : Hello WeApp

新建项目

< 返回

新建项目

AppID

asdf

填写小程序AppID，可在公众平台开发设置页中查看

项目名称

asdf

本地开发目录

F:\wechatApp\test

选择

☒ 在当前目录中创建 quick start 项目

取消

添加项目

- AppID 查看公众开发平台设置查看 (<https://mp.weixin.qq.com>)

基本配置

开发者ID

AppID(应用ID) wx[REDACTED]

AppSecret(应用密钥) [显示密钥](#) [重置](#)



扫码关注
接收重要通知

- 项目名称随意填写
- 本地开发项目：新建一个空的文件夹勾线当前会生成一个项目

打开项目

设置 动作 帮助



Demo



编辑



调试



项目

▼ 文件夹 pages

▼ 文件夹 index

JS index.js

< > index.wxml

{ } index.wxss

▼ 文件夹 logs

JS logs.js

JN logs.json

< > logs.wxml

{ } logs.wxss

▼ 文件夹 main

JS main.js

< > main.wxml

{ } main.wxss

▼ 文件夹 utils

JS util.js

JS app.js

JN app.json

{ } app.wxss



编译

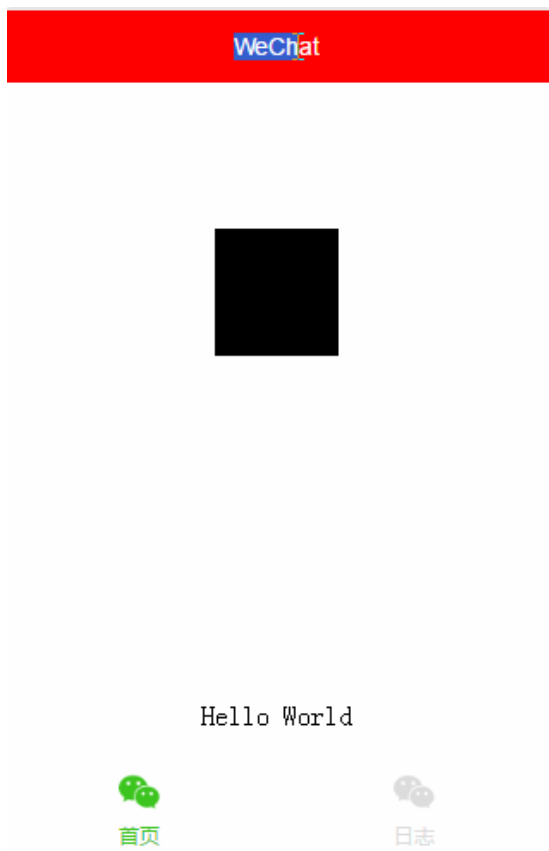


关闭

更详细的说多了也没啥意义看官网的简易教程非常详细！

<https://mp.weixin.qq.com/debug/wxadoc/dev/?t=1474644089807>

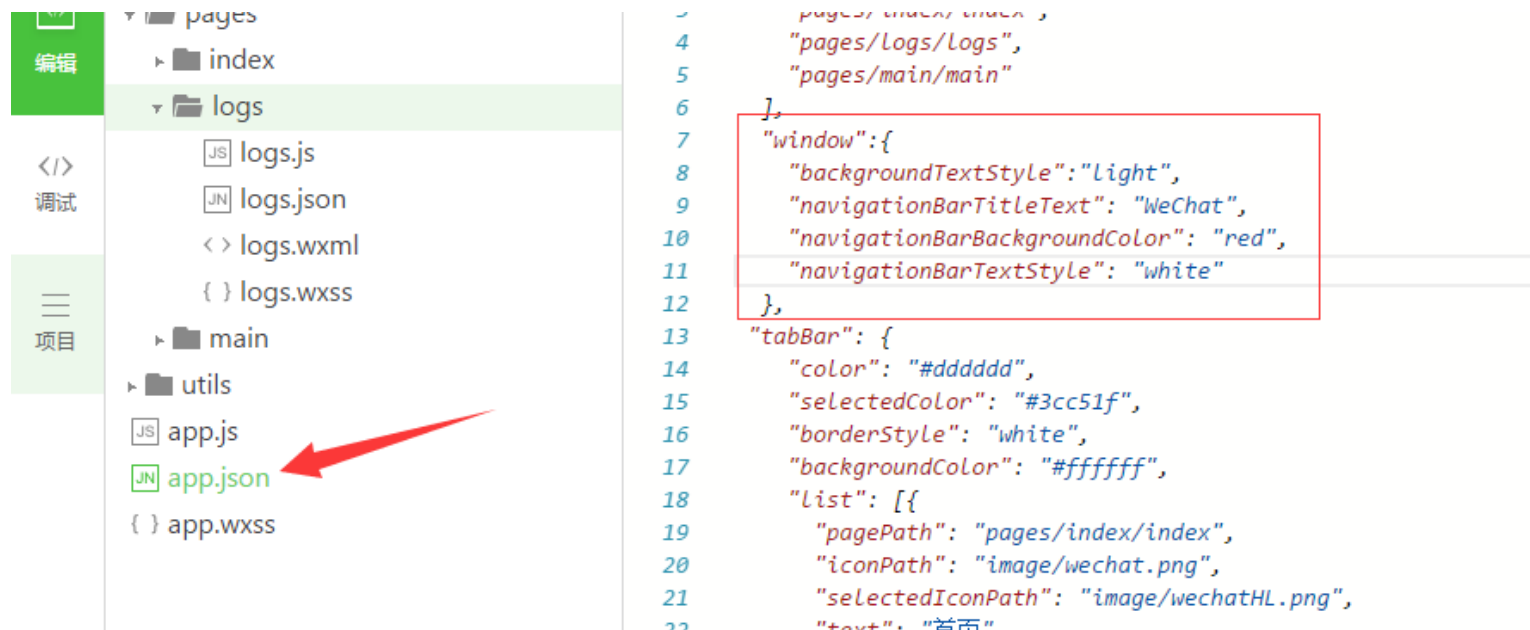
二：window 配置



window 用于设置小程序的状态栏、导航条、标题、窗口背景色。注意在 app.json 中配置的属性会被 window 属性覆盖

| 属性 | 类型 | 默认值 | 描述 |
|------------------------------|----------|---------|------------------------------------|
| navigationBarBackgroundColor | HexColor | #000000 | 导航栏背景颜色，如"#000000" |
| navigationBarTextStyle | String | white | 导航栏标题颜色，仅支持 black/white |
| navigationBarTitleText | String | | 导航栏标题文字内容 |
| backgroundColor | HexColor | #ffffff | 窗口的背景色 |
| backgroundTextStyle | String | dark | 下拉背景字体、loading 图的样式，仅支持 dark/light |
| enablePullDownRefresh | Boolean | false | 是否开启下拉刷新 |

只需在 app.json 配置即可



三：tabBar 底部导航

tabBar 相对而言用的还是比较多的，但是用起来并没有难，在 app.json 中配置下 tabBar 即可，注意 **tabBar 至少需要两个最多五个 Item 选项**

主要属性：

对于 tabBar 整体属性设置：

属性说明：

| 属性 | 类型 | 必填 | 默认值 | 描述 |
|-----------------|----------|----|-------|------------------------------------|
| color | HexColor | 是 | | tab 上的文字默认颜色 |
| selectedColor | HexColor | 是 | | tab 上的文字选中时的颜色 |
| backgroundColor | HexColor | 是 | | tab 的背景色 |
| borderStyle | String | 否 | black | tabbar上边框的颜色， 仅支持 black/white |
| list | Array | 是 | | tab 的列表，详见 list 属性说明，最少2个、最多5个 tab |

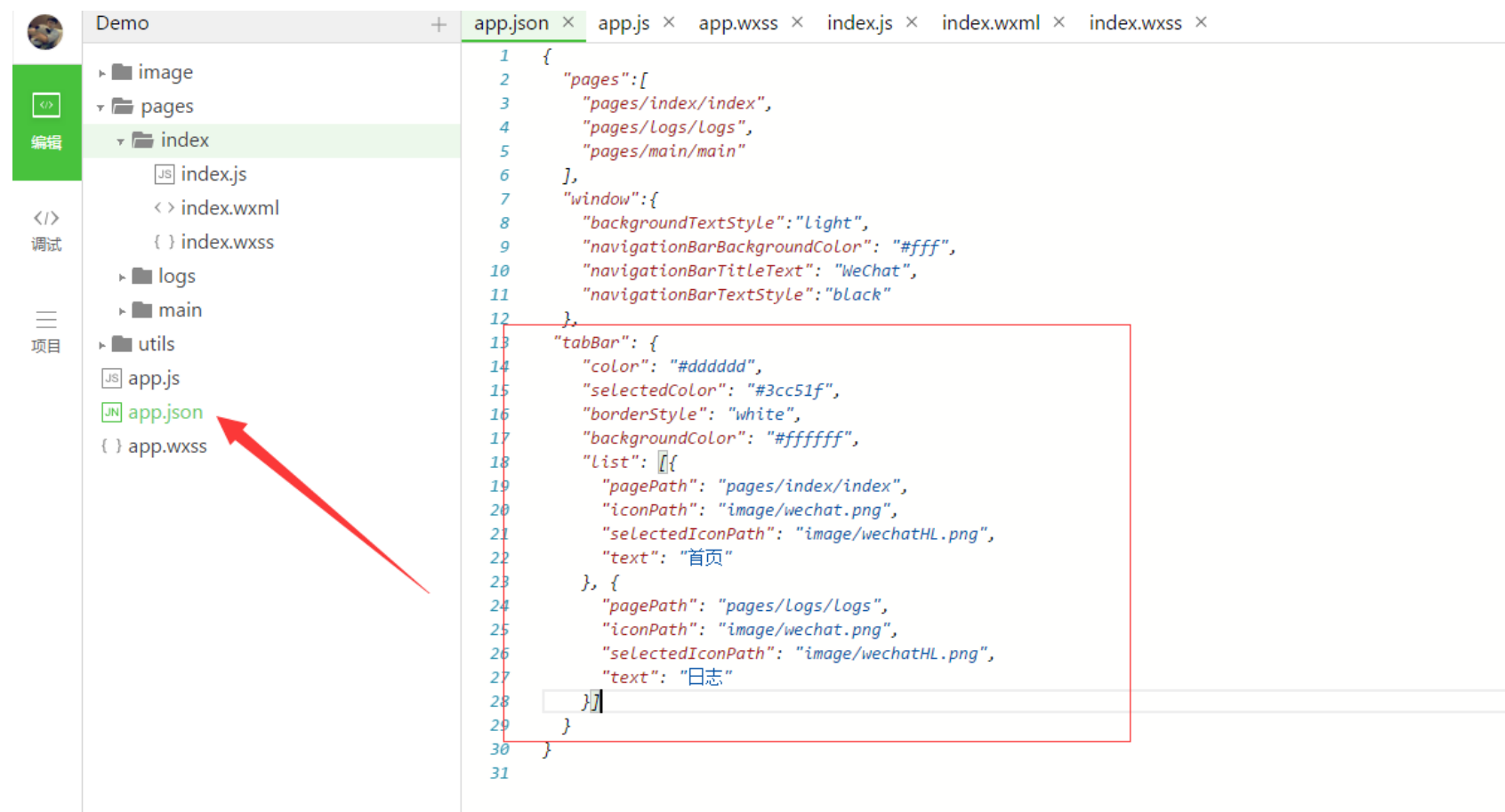
对于 tabBar 中每个 Item 属性设置：

| 属性 | 类型 | 必填 | 说明 |
|------------------|--------|----|-------------------------|
| pagePath | String | 是 | 页面路径，必须在 pages 中先定义 |
| text | String | 是 | tab 上按钮文字 |
| iconPath | String | 是 | 图片路径，icon 大小限制为40kb |
| selectedIconPath | String | 是 | 选中时的图片路径，icon 大小限制为40kb |

下面是官网一张图对 tabBar 描述：



app.json 的配置相对就简单了：



四：应用生命周期

iPhone 6

wifi

编辑

调试

项目

重启

后台

缓存

关闭

WeChat

Hello World

首页 日志 我的

SourcesContent snippetsmain.js x

top1333014643.appservice.open.weixin

pagesutilsappserviceWAService.jsapp.jsasdebug.js(no domain)

```
1 define("pages/main/main.js", function(require, module) {
2   var app = getApp()
3   Page({
4     data: {
5       motto: 'Hello WeApp',
6       userInfo: {}
8     },
9     onButtonTap: function() {
10       wx.navigateTo({
11         url: '../Logs/Logs'
12       })
13     },
14     onLoad: function () {
15       console.log('onLoad')
16       var that = this
17       //登录
18       wx.login({
19         success: function () {
20           wx.getUserInfo({
21             success: function (res) {
22               that.setData({userInfo: res.userInfo})
23               that.update()
24             }
25           })
26         },
27         fail: function (res) {
28           console.log(res)
29         }
30       })
31     }
32   })
33 });require("pages/main/main.js")
```

Line 15, Column 16

ConsoleSearch

top

Preserve log

TypeError: Cannot read property 'MaxRequestConcurrent' of undefined(...)

程序启动我会被调用

程序启动完成我会被调用

onLoad

App() 函数用来注册一个小程序，注意必须在 app.js 中注册，且不能注册多个。

使用方式也跟 **Android** 中的 Application 中初始化一些全局信息以供使用。

方法：

| 属性 | 类型 | 描述 | 触发时机 |
|----------|----------|--|----------------------------------|
| onLaunch | Function | 生命周期函数--监听小程序初始化 | 当小程序初始化完成时，会触发 onLaunch（全局只触发一次） |
| onShow | Function | 生命周期函数--监听小程序显示 | 当小程序启动，或从后台进入前台显示，会触发 onShow |
| onHide | Function | 生命周期函数--监听小程序隐藏 | 当小程序从前台进入后台，会触发 onHide |
| 其他 | Any | 开发者可以添加任意的函数或数据到 Object 参数中，用 <code>this</code> 可以访问 | |

应用生命周期代码：

logs.js × logs.json × app.json × app.wxss × index.wxss × main.js × app.js ×

```
1 //app.js
2 App({
3
4   /**
5    * 程序启动的时候调用，只会调用一次 跟Android中onCreate()方法一样
6    */
7   onLaunch: function () {
8
9     console.log('程序启动我会被调用');
10
11     //调用API从本地缓存中获取数据
12     var logs = wx.getStorageSync('logs') || []
13     logs.unshift(Date.now())
14     wx.setStorageSync('logs', logs)
15   },
16   getUserInfo:function(cb){
17     var that = this;
18     if(this.globalData.userInfo){
19       typeof cb == "function" && cb(this.globalData.userInfo)
20     }else{
21       //调用登录接口
22       wx.login({
23         success: function () {
24           wx.getUserInfo({
25             success: function (res) {
26               that.globalData.userInfo = res.userInfo;
27               typeof cb == "function" && cb(that.globalData.userInfo)
28             }
29           })
30         }
31       });
32     }
33   },
34   globalData:{
35     userInfo:null
36   },
37   /**
38    * 当程序启动完成展示给用户之后，从后台切换到前台会再次调用
39    */
40   onShow: function() {
41     console.log('程序启动完成我会被调用');
42   },
43   /**
44    * 当程序进入后台调用
45    */
46   onHide: function() {
47     console.log('程序进入后台我会被调用');
48   }
49 })
50
```