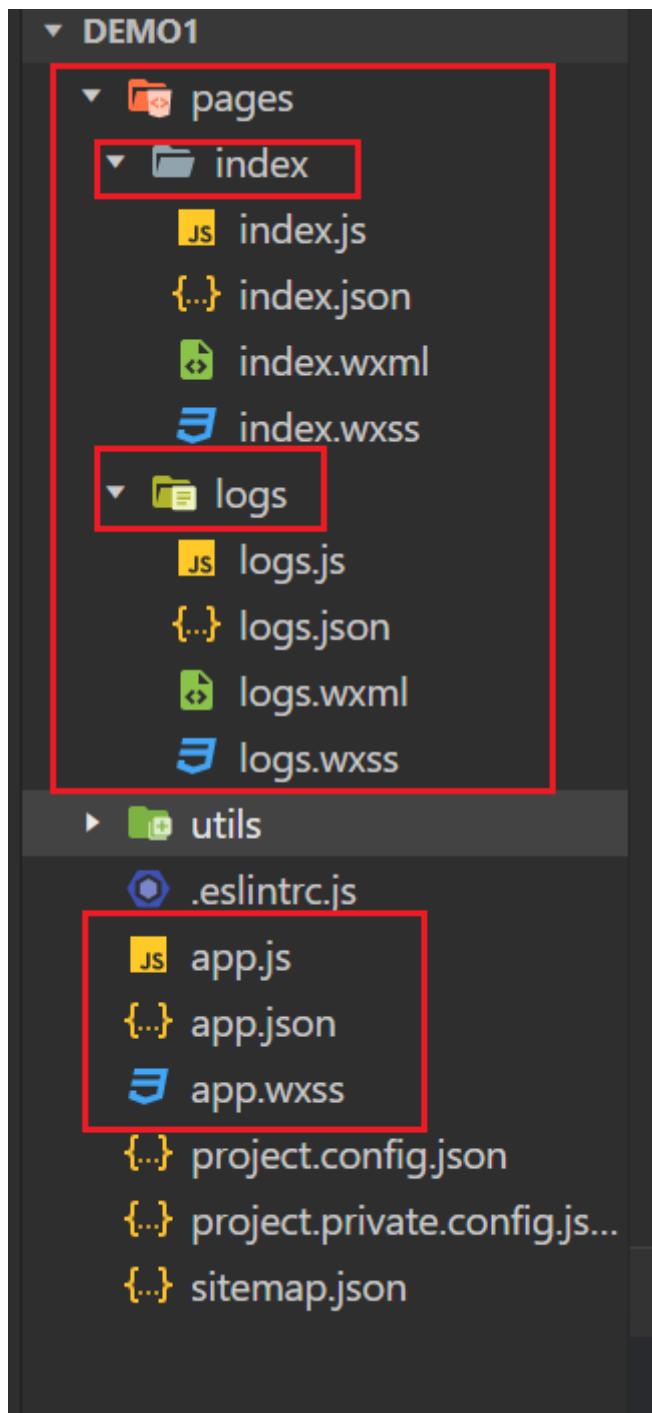


1. 小程序目录结构

小程序包含一个描述整体程序的 `app` 和多个描述各自页面的 `page`

一个小程序主体部分由三个文件组成，必须放在项目的根目录



文件名称	必须	作用
app.js	是	小程序逻辑(小程序项目的入口文件)
app.json	是	小程序公共设置(小程序项目的配置)

文件名称	必须	作用
app.wxss	否	小程序公共样式文件(小程序全局样式)

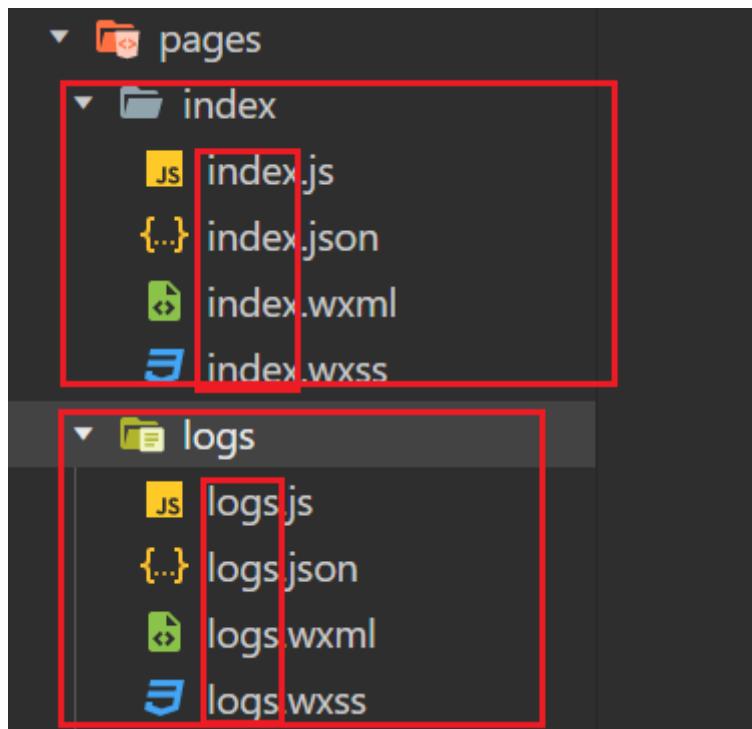
2.pages目录

pages目录 -- 用来存放所有小程序的页面。

- 每个页面包含四个文件
- xx.js: 页面的脚本文件，存放页面的数据、事件、处理函数等。
- xx.json: 当前页面的配置文件，配置窗口的外观、表现等。
- xx.wxml: 页面的模板结构文件。类似html文件
- xx.wxss: 当前页面的样式表文件。类似css文件

文件名称	必须	作用
xxx.js	是	页面逻辑
xxx.wxml	是	页面结构
xxx.json	否	页面配置
xxx.wxss	否	页面样式

注意：为了方便开发者减少配置项，描述页面的四个文件必须具有相同的路径与文件名



3.全局配置 app.json

app.json: 当前程序的全局配置文件，包含了小程序所有页面的路径、窗口外观、界面表现、底部tab、设置网络超时时间等

---- 注意

----- app.json中不能添加任何注释，否则会报错

----- 字符串用双引号引起起来。

```
app.json
{
  "pages": [
    "pages/index/index",
    "pages/logs/logs"
  ],
  "window": {
    "backgroundTextStyle": "light",
    "navigationBarBackgroundColor": "#fff",
    "navigationBarTitleText": "Weixin",
    "navigationBarTextStyle": "black"
  },
  "style": "v2",
  "sitemapLocation": "sitemap.json"
}
```

The screenshot shows the file structure of a WeChat小程序 project. The 'app.json' file is open in the editor, displaying its contents. Several parts of the JSON object are highlighted with green boxes: 'pages', 'window', and 'sitemapLocation'. The 'pages' array contains two page paths: 'pages/index/index' and 'pages/logs/logs'. The 'window' object contains properties for the background text style, navigation bar background color, title text, and text style. The 'style' key is set to 'v2', and the 'sitemapLocation' key is set to 'sitemap.json'. The left sidebar shows the project structure with files like index.js, index.json, index.wxml, index.wxss, logs.js, logs.json, logs.wxml, logs.wxss, .eslintrc.js, app.js, and sitemap.json.

上图中我们看到app.json中实际保存的是小程序的主配置项属性列表，app.json中的属性设置有以下几项：

微信现已开放小程序内搜索，开发者可以通过 `sitemap.json` 配置，或者管理后台页面收录开关来配置其小程序页面是否允许微信索引；当开发者允许微信索引时，微信会通过爬虫的形式，为小程序的页面内容建立索引。当用户的搜索词条触发该索引时，小程序的页面将可能展示在搜索结果中。

`pages`: 用来记录当前小程序所有页面的路径

`window`: 全局定义小程序的所有页面的背景色、文字颜色等。

`style`: 全局定义小程序组件所使用的样式版本。v2代表全新的样式

`sitemapLocation`: `sitemap`的存放路径。

上述均为默认的，除了上述属性外还有以下常用属性：

`tabBar`: 设置底部tab的表现

`networkTimeout`: 设置网络超时时间

`debug`: 设置是否开启debug

属性	类型	必填	描述
pages	String Array	是	设置页面路径
window	Object	否	设置默认页面的窗口表现
tabBar	Object	否	设置底部 tab 的表现
networkTimeout	Object	否	设置网络超时时间
debug	Boolean	否	设置是否开启 debug 模式

1. **enterPagePath** 【string类型】：小程序默认启动首页
2. **pages** 【string数组】：指定小程序的组成页面
3. **window** 【object对象】：指定小程序的状态栏、导航条、标题、窗口背景颜色
4. **tabBar** 【object对象】：指定小程序的底部导航栏
5. **functionalPages** 【Boolean类型】：指定小程序是否启用插件功能页
6. **plugins** 【object对象】：指定小程序使用的插件
7. **resizable** 【Boolean类型】：指定小程序是否支持改变大小（PC）/屏幕旋转（iPad）
8. **usingComponents** 【object类型】：指定小程序全局自定义组件的配置
9. **permission** 【object类型】：指定小程序接口权限设置（待定）
10. **sitemapLocation** 【string类型】：指定sitemap.json配置文件的位置，该文件用于设置小程序被微信爬虫索引时应遵循的规则列表
11. **useExtendedLib** 【object对象】：指定小程序需要引用的扩展库
12. **darkmode** 【Boolean类型】：指定小程序是否根据系统主题色展示不同样式
13. **themLocation** 【string类型】：指定theme.json配置文件的位置，该文件用于设置小程序对主题改变时相关颜色的改变规则

参考

```
{
  "pages": [
    "pages/index/index",
    "pages/logs/index"
  ],
  "window": {
    "navigationBarTitleText": "Demo"
  },
  "tabBar": {
    "list": [
      {
        "pagePath": "pages/index/index",
        "text": "首页"
      },
      {
        "pagePath": "pages/logs/logs",
        "text": "日志"
      }
    ]
  },
  "networkTimeout": {
    "request": 10000,
    "downloadFile": 10000
  },
  "debug": true
}
```

1.pages

接受一个数组，每一项都是字符串，来指定小程序由哪些页面组成。

每一项代表对应页面的【路径+文件名】信息

---- 数组的第一项代表小程序的初始页面。小程序中新增/减少页面，都需要对 `pages` 数组进行修改。

---- 文件名不需要写文件后缀，因为框架会自动去寻找路径`.json`,`.js`,`.wxm1`,`.wxss`的四个文件进行整合。

注意：小程序每新增一个页面，相应的在`pages`中就需要增加多一个配置页面记录；反之亦然。

如开发目录为：

```
|-- app.js
|-- app.json
|-- app.wxss
|-- pages
|   |-- index
|   |   |-- index.wxml
|   |   |-- index.js
|   |   |-- index.json
|   |   |-- index.wxss
|   |-- logs
|       |-- logs.wxml
|       |-- logs.js
|-- utils
```

则需要在 `app.json` 中写

```
{
  "pages": ["pages/index/index", "pages/logs/logs"]
}
```

2.windows

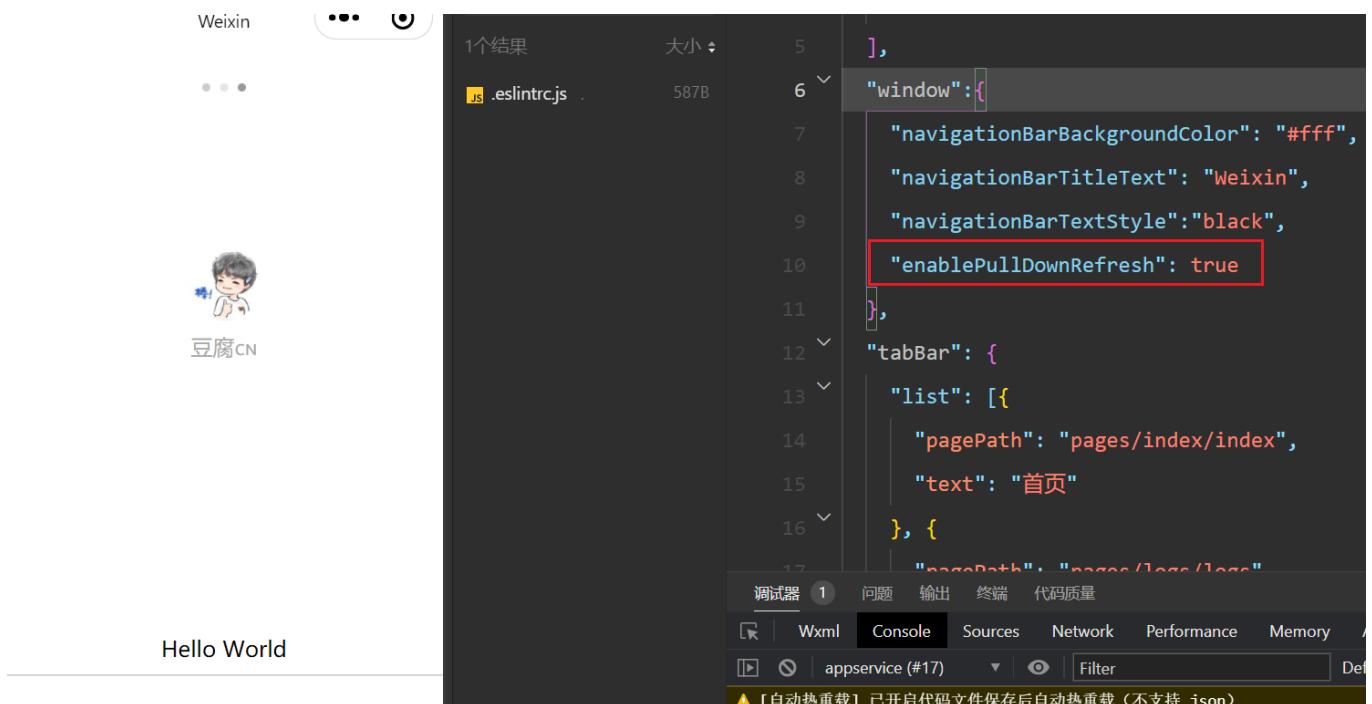
用于设置小程序的状态栏、导航条、标题、窗口背景色。

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，详见 页面相关事件处理函数 。

例如：

```
{
  "window":{
    "navigationBarBackgroundColor": "#ffffff",
    "navigationBarTextStyle": "black",
    "navigationBarTitleText": "微信接口功能演示",
    "backgroundColor": "#eeeeee",
    "backgroundTextStyle": "light"
  }
}
```

下拉刷新

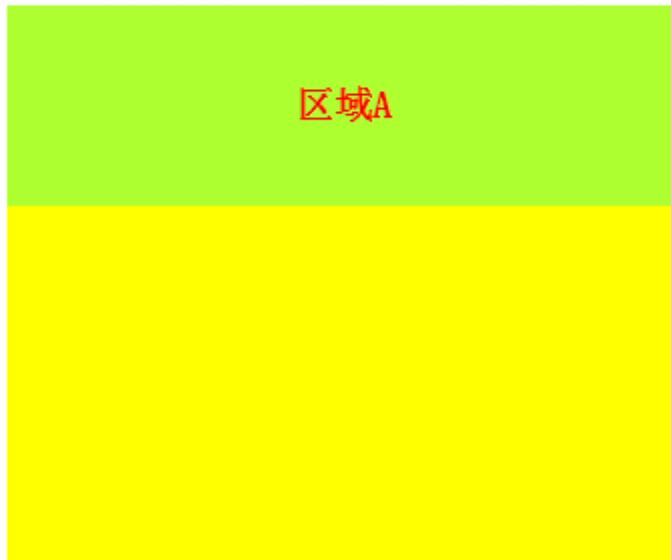
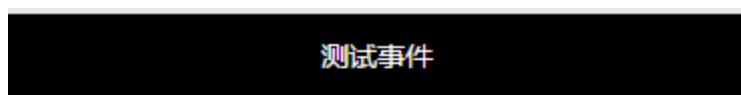


刷新三个点



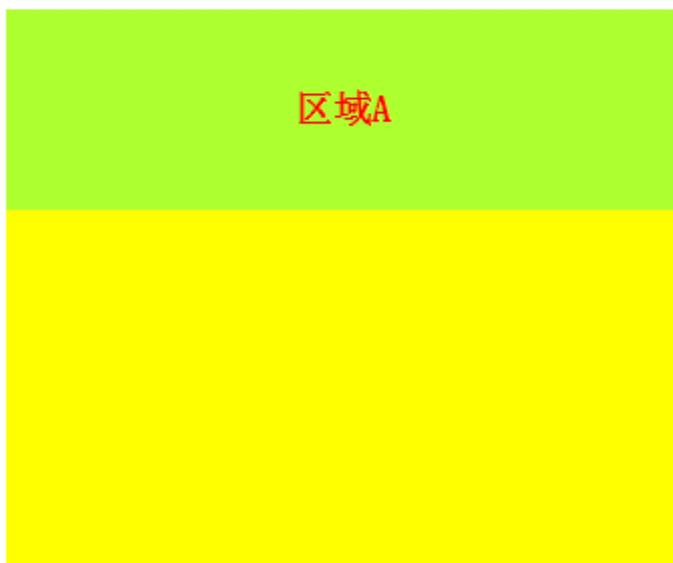
```
无依赖文件 搜索
1个结果 大小 587B
js .eslintrc.js 587B
4 "pages/logs/logs"
5 ],
6 "window": [
7   "backgroundTextStyle": "dark",
8   "navigationBarBackgroundColor": "#fff",
9   "navigationBarTitleText": "Weixin",
10  "navigationBarTextStyle": "black",
11  "enablePullDownRefresh": true
12 },
13 "tabBar": {
14   "list": [
15     {
16       "pagePath": "pages/index/index",
17       "text": "首页"
18     }
19   ]
20 }
21 调试器 1 问题 输出 终端 代码质量
22 Wxml Console Sources Network Performance Memory AppData Storage Security
23 [appservice (#16)] Default levels ▾
24 ▲ [自动热重载] 已开启代码文件保存后自动热重载 (不支持 json)
```

backgroundTextStyle -- light



backgroundTextStyle -- dark --- 灰色动画

测试事件



刷新下拉框颜色

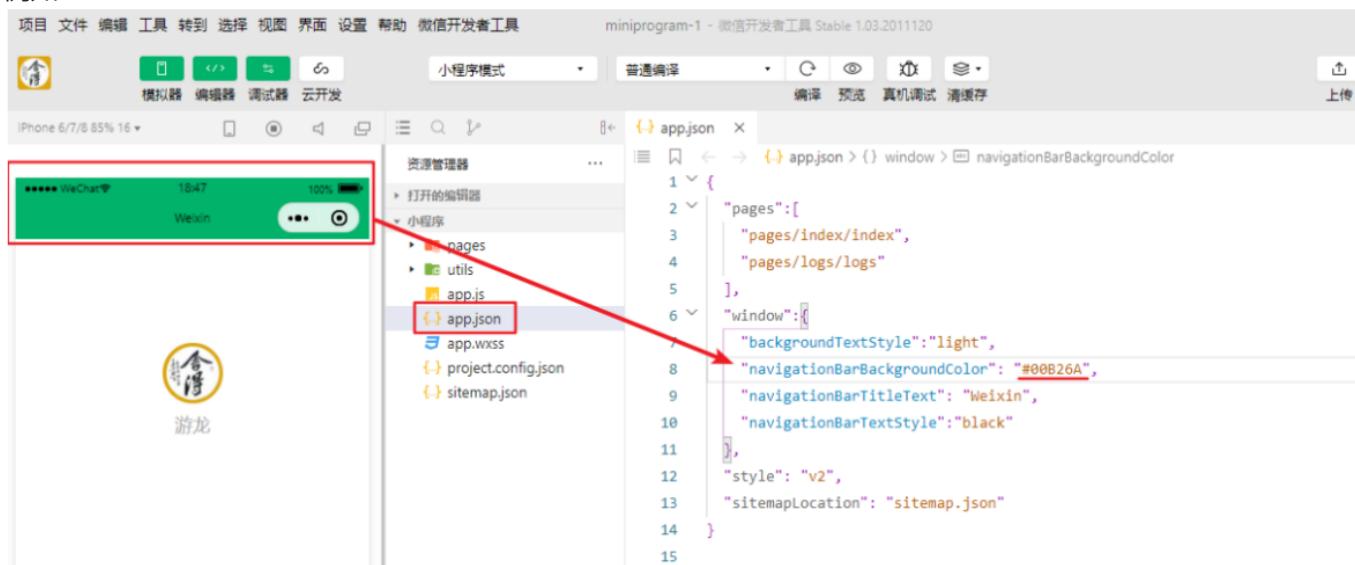
The screenshot shows the WeChat developer tools interface. On the left is a preview window of a WeChat page titled "Hello World". The page has a blue header bar with a navigation icon and the text "Weixin". Below the header is a white area with a user profile picture and the text "豆腐CN". At the bottom of the page is a navigation bar with tabs for "首页" and "日志". On the right is a code editor window displaying a file named ".eslintrc.js". The code defines a "window" object with properties like "navigationBarBackgroundColor" set to "#0094ff". The "navigationBarTitleText" is set to "Weixin". The "navigationBarTextStyle" is "black". The "enablePullDownRefresh" property is set to true. A "tabBar" object is defined with a "list" containing a single item where the "pagePath" is "pages/index/index". The code editor has syntax highlighting and line numbers. Below the code editor is a toolbar with tabs for "调试器" (Debugger), "Console", "Sources", "Network", "Performance", "Memory", "AppData", "Storage", and "Se". The "Console" tab is selected. At the bottom of the interface is a status bar with a warning message: "⚠️ [自动热重载] 已开启代码文件保存后自动热重载 (不支持 json)".

```
4      "pages/logs/logs"
5    ],
6    "window": {
7      "backgroundTextStyle": "dark",
8      "backgroundColor": "#0094ff",
9      "navigationBarBackgroundColor": "#fff",
10     "navigationBarTitleText": "Weixin",
11     "navigationBarTextStyle": "black",
12     "enablePullDownRefresh": true
13   },
14   "tabBar": {
15     "list": [
16       {
17         "pagePath": "pages/index/index",
18         "+--+",
19         "首页"
20       }
21     ]
22   }
23 }
```

// 注: HexColor (十六进制颜色值) , 如 "#ff00ff"

属性	类型	默认值	描述	最低版本
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	
navigationStyle	String	default	导航栏样式, 仅支持 default/custom.custom 模式可自定义导航栏, 只保留右上角胶囊状的按键	微信版本6.6.0
backgroundColor	HexColor	#ffffff	窗口的背景色	
backgroundTextStyle	String	dark	下拉 loading 的样式, 仅支持dark/light	
backgroundColorTop	String	#ffffff	顶部窗口的背景色, 仅 IOS支持	微信版本6.5.16
backgroundColorBottom	String	#ffffff	底部窗口的背景色, 仅 IOS支持	微信版本6.5.16
enablePullDownRefresh	Boolean	false	是否开启下拉刷新, 详见页面相关事件处理 函数	
onReachBottomDistance	Number	50	页面上拉触底事件触发时距页面底部距离, 单位为px	

例如:



3.tabBar

如果我们的小程序是一个多 tabBar 应用

(客户端窗口的底部或顶部有 tabBar 样式可以切换页面),

那么我们可以通过 tabBar 配置项指定 tabBar 样式的对应页面

注: 通过页面跳转 (`wx.navigateTo`) 或者页面重定向 (`wx.redirectTo`) 所到达的页面, 即使它是定义在 tabBar 配置中的页面, 也不会显示底部的 tabBar 样式。

注: tabBar 是一个数组, 只能配置最少2个、最多5个 tabBar, tabBar 按数组的顺序排序。

注: 当设置 position 为 top 时, 将不会显示 icon。

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
position	String	否	bottom	可选值 bottom、top

其中 list 接受一个数组，数组中的每个项都是一个对象，其属性值如下：

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，建议尺寸为 81px * 81px
selectedIconPath	String	是	选中时的图片路径，icon 大小限制为40kb，建议尺寸为 81px * 81px

例如：

打开的编辑器

WECHATPROJECT

- icons
 - cart-o.png
 - cart.png
 - category-o.png
 - category.png
 - home-o.png
 - home.png
 - my-o.png
 - my.png
- pages
 - index
- logs
 - logs.js
 - logs.json
 - logs.wxml
 - logs.wxss
- utils
- app.js
- app.json
- app.wxss

```

10   "navigationBarTextStyle": "black"
11 },
12   "tabBar": {
13     "color": "#00b26a",
14     "selectedColor": "#000000",
15     "backgroundColor": "#ce9083",
16     "list": [
17       {
18         "pagePath": "pages/index/index",
19         "text": "首页",
20         "iconPath": "/icons/home.png",
21         "selectedIconPath": "/icons/home-o.png"
22       },
23       {
24         "pagePath": "pages/logs/logs",
25         "text": "日志",
26         "iconPath": "/icons/my.png",
27         "selectedIconPath": "/icons/my-o.png"
28       }
29     ],
30     "style": "v2",
31     "sitemapLocation": "sitemap.json"
32   }
33 
```

表单组件

操作反馈

backgroundColor

导航

selectedIconPath

具体组件

borderStyle

iconPath

组件

selectedColor

接口

color

4.networkTimeout

可以设置各种网络请求的超时时间。

网络开发文档

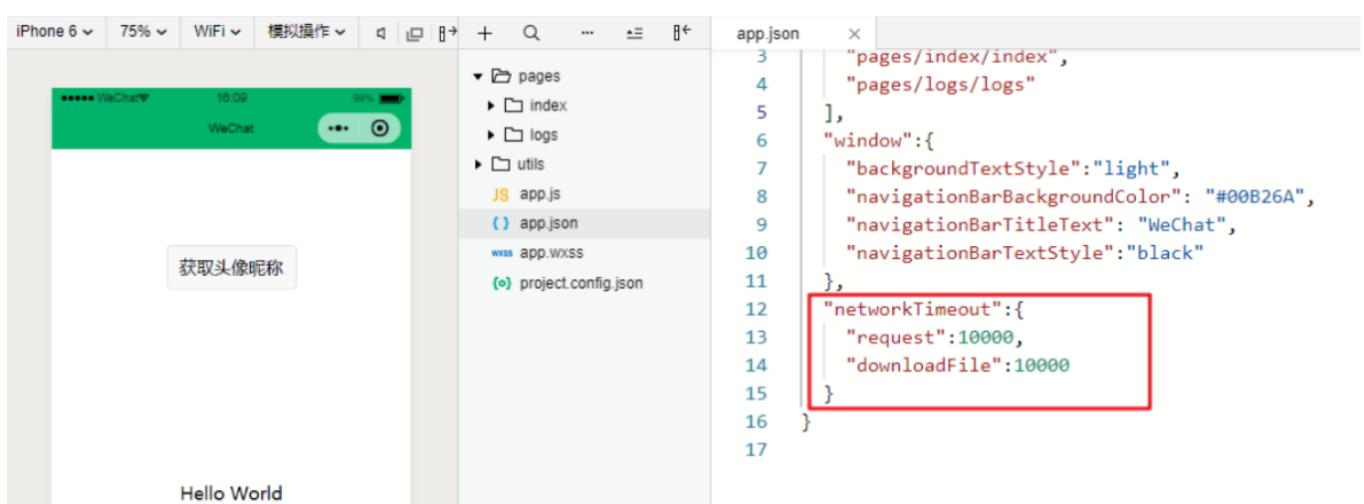
<https://developers.weixin.qq.com/miniprogram/dev/framework/ability/network.html>

wx.connectSocket --- 创建一个 WebSocket 连接。

<https://developers.weixin.qq.com/miniprogram/dev/api/network/websocket/wx.connectSocket.html>

属性	类型	必填	说明
request	Number	否	wx.request的超时时间，单位毫秒，默认为：60000
connectSocket	Number	否	wx.connectSocket的超时时间，单位毫秒，默认为：60000
uploadFile	Number	否	wx.uploadFile的超时时间，单位毫秒，默认为：60000
downloadFile	Number	否	wx.downloadFile的超时时间，单位毫秒，默认为：60000

注意：上线之前一定要设置一下超时时间，不然小程序有可能出现下列情况(请求数据获取失败)。



5.debug

可以在开发者工具中开启 `debug` 模式，在开发者工具的控制台面板，调试信息以 `info` 的形式给出，其信息有Page的注册，页面路由，数据更新，事件触发。可以帮助开发者快速定位一些常见的问题

在开发阶段，建议打开 `debug`。上线时请关闭此选项，设置为 `false`。

```

{
  "pages": [
    "pages/index/index",
    "pages/logs/index"
  ],
  "window": {
    "navigationBarTitleText": "Demo"
  },
  "tabBar": {
    "list": [
      {
        "pagePath": "pages/index/index",
        "text": "首页"
      },
      {
        "pagePath": "pages/logs/index",
        "text": "日志"
      }
    ]
  }
}
  
```

```
        "text": "日志"
    }
],
},
"networkTimeout": {
    "request": 10000,
    "downloadFile": 10000
},
"debug": true
}
```

4. 页面配置 xx.json

每个小程序页面也可以使用同名 *.json 文件来对本页面的窗口表现进行配置，页面中配置项会覆盖app.json 的 window 中相同的配置项。

用于设置小程序的状态栏、导航条、标题、窗口背景色。

开发文档：

<https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/page.html>

静态配置参考

```
{
    "navigationBarBackgroundColor": "#ffffff",
    "navigationBarTextStyle": "black",
    "navigationBarTitleText": "微信接口功能演示",
    "backgroundColor": "#eeeeee",
    "backgroundTextStyle": "light"
}
```



静态配置参考

使用微信提供的API接口

语法: `wx.setNavigationBarTitle(Object object)`

The screenshot shows the WeChat Mini Program development environment. On the left is the file tree:

- 资源管理器
- 打开的编辑器
- 优雅谷
 - pages
 - icon
 - index
 - logs
 - test
 - test.js
 - test.json
 - test.wxml
 - test.wxss
- utils
 - util.js
 - app.js
 - app.json
 - app.wxss

The code editor shows `test.js` with the following content:

```
10 /**
11 * 生命周期函数--监听页面加载
12 */
13 onLoad: function (options) {
14     // 动态设置当前页面标题
15     wx.setNavigationBarTitle({
16         "title": "我们是最认真..."
17     })
18 }
19 /**
20 * 生命周期函数--监听页面初次渲染完成
21 */
22 onReady: function () {
23 }
```

A red box highlights the `wx.setNavigationBarTitle` call.

==注意：调用微信提供的API设置当前页面标题的级别比*.json文件高

5.sitemap配置

微信现已开放小程序内搜索，开发者可以通过 `sitemap.json` 配置，或者管理后台页面收录开关来配置其小程序页面是否允许微信索引；当开发者允许微信索引时，微信会通过爬虫的形式，为小程序的页面内容建立索引。当用户的搜索词条触发该索引时，小程序的页面将可能展示在搜索结果中。

开发文档：

<https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/sitemap.html>

注意：`app.json`中`sitemapLocation`指向的就是`sitemap.json`

`sitemap.json`专门配置`sitemap`

参考官方文档：

`{"action": "allow", "page": "*"} 是优先级最低的默认规则，未显式指明 "disallow" 的都默认被索引`