小程序开发文档

<https://developers.weixin.qq.com/miniprogram/dev/quickstart/basic/code.html>

小程序特点

1. 位于微信公众平台
2. 无需进行下载

注意：微信 项目目录不允许有中文

微信小程序-目录结构与文件类型

wx/pages 项目所有组件保存目录

index.wxml 首页：严格的一组xml标签

app.wxss 全局样式（单位 rpx;@import “ws.wxss”）

app.js 入口程序,全局变量存储globalData,全局app实例，其他页面通过getApp()调用

app.json 项目配置文件

sitemap.json 配置小程序及页面是否能被索引

微信小程序-目录结构与文件类型-wxml

类似于html，但是比html严格：标签一定成对，属性一定要有“”

常用标签

<view></view> == <div></div>

<text></text> == <span></span>

<image></image> == <img src=””>

<navigator></navigator> == <a>

<form></form>

img标签<image src=””></image>

有默认大小

图片路径:绝对路径、相对路径、网络绝对路径

常用属性

lazy-load ：true/false

mode： 图片加载策略

aspectFill 显示图片等比例缩放 以短边为准

aspectFit 显示图片等比例缩放 以长边为准

text标签

属性：

selectable : true/false 是否可选择 复制

decode： true/false 是否对文本进行编码

小程序在运行时有一个自动化操作，在运行自动将wxss;js加载到当前wxml无需引入

wxss文件

rpx写像素

font-size 写px

@import “” 引入其他样式文件

js文件

小程序为每一个组件对应创建对象和.js文件

Page() 当前组件 对象

Page({})

有ECMAScript

没有DOM/BOM

小程序顶级对象wx == window

data与vue相近双花括号模板显示

this.setData({ 修改data中的数据

msg: now

})

获取数据 let x = this.data.xxx;

小程序组件生命周期

onLoad: 组件加载完成触发一次 发送ajax请求，获取其他组件传递的参数onload(options){} options为参数

onReady: 组件初始渲染完成触发一次

onShow: 组件显示触发多次、将组件切换前后台操作

onHide: 组件隐藏或者切换后台执行

onUnload: 组件卸载触发（将组件移出内存）一次

页面跳转 redirectTo({url: ‘’})

组件中特殊事件

onPullDownRefresh 监听下拉操作#默认小程序禁止用户下拉操作

onReachBottom 监听上拉操作

onPageScroll：funciton(opt){} 监听用户滑动操作 opt滑动的位置

解决：更改全局配置 app.json “window”:{enablePullDownRefresh: true};

全局配置项app.json

pages:array 配置组件路径

window 窗口样式

"backgroundTextStyle": "light", 背景样式

"navigationBarBackgroundColor": "#fff",导航条背景色

"navigationBarTitleText": "WeChat", 导航条文字

"navigationBarTextStyle": "black" 导航条文本颜色

tabbar 底部导航栏

tabbar:{

list:[

{

pagePath:组件路径，

text:”文字”，

iconPath: “图片路径”，

selectedIconPath: “”

},

至少2个

]

}

小程序模板 数据显示

小程序动态获取data,

if/else/elif

<view wx:if=”{{判断条件}}”>xxx</view>

for

<view wx:for={{数组对象}}>默认下 为item与 index

<view wx:for-item=”当前元素”wx:for-index=”下标”>

微信小程序事件

pc事件：依靠鼠标

移动端事件：依靠手指

原生DOM通用事件

touchstart 当手指触碰屏幕时发生，不管几个手

touchmove 当手指在屏幕滑动时连续触发

touchend 当手指离开屏幕时触发

zepto.js 移动端js库== jquery

触碰函数

tap 按一下

longtap 长按

swipe 滑动

swipeLeft/Right/Up/Down 上下左右滑动 注意事项：禁止屏幕滑动：css \*{touch-action: none}

微信小程序事件

touchstart 手指触摸动作开始

touchmove 手指触摸后移动

touchcancel 手指触摸动作被打断，如来电提醒，弹窗

touchend 手指触摸动作结束

tap 手指触摸后马上离开

longpress 手指触摸后，超过350ms再离开，如果指定了事件回调函数并触发了这个事件，tap事件将不被触发 1.5.0

longtap 手指触摸后，超过350ms再离开（推荐使用longpress事件代替）

transitionend 会在 WXSS transition 或 wx.createAnimation 动画结束后触发

animationstart 会在一个 WXSS animation 动画开始时触发

animationiteration 会在一个 WXSS animation 一次迭代结束时触发

animationend 会在一个 WXSS animation 动画完成时触发

touchforcechange 在支持 3D Touch 的 iPhone 设备，重按时会触发

小程序事件绑定

<view bind：事件名=”函数”>支持事件冒泡</view>

<view catch：事件名=”函数”>不支持事件冒泡</view>

函数写在 Page对象中

小程序事件捕获

<view capture-bind：事件名=”函数”>支持事件冒泡</view>

<view capture-catch：事件名=”函数”>不支持事件冒泡</view>

微信小程序事件对象 event

type:’’事件名称

currentTarget: 当前元素

target: 触发事件元素

timeStamp: 当组件打开到触发事件经过毫秒

元素与处理函数传递值

<view bindtap=”handle” data-name=””></view>

e.target.dataset.name;

小程序的基础组件

<组件名 id class style hidden>组件内容</组件名>

组件之间跳转

标签

<navigator url=”组件路径”>文字</navigator>

wx.redirectTo({url:”组件路径”})；卸载上一组件，不能跳转tabbar中定义组件

wx.navigateTo（{url:”组件路径”}）；不会卸载上一组件，也不能跳转tabbar中定义的组件

wx.navigateBack() 调用左上角返回

wx.switchTab() 调用tab键切换，只能打开tabbar页面

tabbar中是不能通过标签方法跳转

wx.reLaunch({url:’’}) 不删前一个组件，可以跳入tabbar

程序与标签中指定组件路径需要添加 /

配置中不用

小程序传参

跳转传参： url:”/xxx/xxx/?xx=xx&xx=xx”

参数接受： onLoad:function(option){

option.xxx

}

小程序ajax

wx.request({});小程序发送异步请求

url 请求服务器地址

method post get 默认get

data 请求参数

success 请求成功回调函数

fail 请求失败回调函数

complete 请求都回调的函数

127.0.0.1小程序默认出错

小程序发起请求必须

合法域名

服务器加密处理

轮播图

<swiper>

<swiper-item>

<image src=””></image>

</swiper-item>

</swiper>

swiper属性：

autoplay 自动播放

interval 间隔时间

indicator-dots 指示点

vertical 是否是纵向

小程序加载动画

wx.showLoading({itle:’’})加载

w.hideloading();移出

小程序表单

<form bindsubmit=”checkForm”>

<input >

<button form-type=”submit”>添加商品</button>

小程序提示消息

wx.showToast({

title: “商品格式不正确”，

icon: “none”

});

wx.hideToast();

小程序全局贡献数据

app.js

App({

globalData: { 数据：值}

})

加载全局组件 const app = getApp();

依据全局组件获取全局数据

值 = app.globalData.数据名

依据全局修改全局数据

app.globalData.数据属性名 = 值；

小程序多媒体

小程序后台播放器可以用于播放音乐，小程序要求同一时刻只能有一个背景音频在播放，当有其他小程序占用播放器来播放音乐，当前音频停止。

方法与事件

播放背景音乐

wx.playBackgroundAudio({

datalist: ‘’,地址

title: 标题

coverImUrl 封面

})

暂停播放背景音乐

wx.pauseBackgroundAudio();

停止播放背景音乐

wx.stopBackgroundAudio();

播放背景音乐触发事件

wx.onBackgroundAudioPlay

暂停背景音乐触发事件

wx.onBackgroundAudioPause

小程序上传文件

wx.chooseImage({

count: 最多可以选中图片总数 默认9

sourceType:[‘camera’, ‘album’]指定图片来源

success: res => {} 选择成功

res.tempFilePaths 本地图片路径列表

});选择图片：本地相机或者相册

wx.uploadFile({

url： 上传的服务器路径

filePath 本地图片路径

name 上传图片参数名称

header Content-Type：“multipart/form-data”

success

}); 选中图片上传服务器

js模块化开发

用module.exports.名字 = 名字 ；暴露接口

需要引入接口的模块

const xxx = require(‘’);

小程序API规则

on开头的API监听事件触发 切每个函数都保函回调与回调值wx.onxx(res => {})

Sync结尾为同步API 会抛出错误 有函数返回值

其他类型API 都包含 success回调 成功时回调 ，fail回调 失败时回调， complete回调 调用该API完成时回调 切保函三个参数 errMsg， errCode, 其他参数

小程序wxml模板化

<template name = ‘模板名’>

模板内容

</template>

模板调用

<template is = ‘模板名’data=’所需要使用的数据’></template>

小程序样式表导入

@import ‘相对路径’；

建议小程序在必要时使用 wx.onMemoryWarning 监听内存告警事件，进行必要的内存清理。

如果需要马上应用最新版本，可以使用 wx.getUpdateManager API 进行处理。

小程序自定义组件

json文件中 component:true

wxml中 定义组件模板骨架可以定义子模板<slot></slot>只能有一个

需要多个slot需要到component js中声明 multipleSlots: true ，并且提供不同name

在自定义组件的 js 文件中，需要使用 Component 来注册组件，并提供组 件的属性定义、内部数据和自定义方法。

自定义组件的使用"usingComponents": {

"组件的应用名": "组件的路径 "

引用的wxml调用模板名模板内提供子模板输入，在父组件中prop-a,prop-b进行传值,子模板的slot的调用为附模板中的view，多个slot可以view添加对应name进行确认

Component({

properties: {},数据

methods: {

onTap() { 定义的方法

this.triggerEvent('myevent', detail对象，提供给事件监听函数, 触发事件的选项) 用此函数触发事件

默认事件触发类型

bubbles Boolean 否 false 事件是否冒泡

composed Boolean 否 false 事件是否可以穿越组件边界，为false时，事件将只能在引用组件的节点树上触发，不进入其他任何组件内部

capturePhase Boolean 否 false 事件是否拥有捕获阶段

}

}

组件生命周期

lifetimes: {

created在组件实例刚刚被创建时执行

attached 在组件实例进入页面节点树时执行

ready 在件在视图层布局完成后执行

moved 在组件实例被移动到节点树另一个位置时执行

detached 在组件实例被从页面节点树移除时执行

error Object Error 每当组件方法抛出错误时执行},

组件所在页面生命周期

pageLifetimes: {

show() {

// 页面被展示

},

hide() {

// 页面被隐藏

},

resize(size) {

// 页面尺寸变化

}

}

behaviors 是用于组件间代码共享的特性在开头引入const myBehavior = require('my-behavior')

behaviors: [myBehavior],

父子自定义组件通信

必须在两个组件定义中都加入relations定义，否则不会生效。

relations

type 目标组件的相对关系，可选的值为 parent 、 child 、 ancestor 、 descendant

linked 关系生命周期函数，当关系被建立在页面节点树中时触发，触发时机在组件attached生命周期之后

linkChanged 关系生命周期函数，当关系在页面节点树中发生改变时触发，触发时机在组件moved生命周期之后

unlinked 关系生命周期函数，当关系脱离页面节点树时触发，触发时机在组件detached生命周期之后

target 如果这一项被设置，则它表示关联的目标节点所应具有的behavior，所有拥有这一behavior的组件节点都会被关联

数据监听

observers: {

'监听数据 \*\*监听数据通配符': function (numberA, numberB) {

监听逻辑

this.setData({

sum: numberA + numberB

})

}

抽象节点可以指定一个默认组件，当具体组件未被指定时，将创建默认组件的实例。默认组件可以在 componentGenerics 字段中指定：

"componentGenerics": {

"selectable": {

"default": "path/to/default/component"

}

}

})

“内网穿透”

<https://ngrok.com>

ngrok.exe http 3000

微信小程序官方脚手架

-miniprogram\_dev <--生产环境运行npm run dev所生产的

-|-components <--生产环境下生产的组件

-|-pages <--文件位置

-src <--开发下源码位置

-|-组件开发位置

-tools <-- 转化工具文件夹

-|-demo <--转化根文件夹

----|-pages <--

----|-app.js <---唯一app主程序

----|-app.json <---主程序配置文件

----|-app.wxss

----|-package.json

----|-package.config.json