

小程序系统API

讲师: 王红元
微博: coderwhy



更多学习资源



老师答疑交流



网络请求 – 基本使用

■ 微信提供了专属的API接口,用于网络请求: `wx.request(Object object)`

属性	类型	默认值	必填	说明	最低版本
url	string		是	开发者服务器接口地址	
data	string/object/ArrayBuffer		否	请求的参数	
header	Object		否	设置请求的 header, header 中不能设置 Referer。 content-type 默认为 application/json	
method	string	GET	否	HTTP 请求方法	
dataType	string	json	否	返回的数据格式	
responseType	string	text	否	响应的数据类型	1.7.0
success	function		否	接口调用成功的回调函数	
fail	function		否	接口调用失败的回调函数	
complete	function		否	接口调用结束的回调函数（调用成功、失败都会执行）	

■ 比较关键的几个属性解析:

- url: 必传, 不然请求什么.
- data: 请求参数
- method: 请求的方式
- success: 成功时的回调
- fail: 失败时的回调



网络请求 – 代码演练

// 1.没有参数的网络请求

```
wx.request({  
  url: 'http://123.207.32.32:8000/recommend',  
  success: function(res) {  
    console.log(res)  
  }  
})
```

// 2.有参数的网络请求

```
wx.request({  
  url: 'http://123.207.32.32:8000/home/data',  
  data: {  
    type: 'sell',  
    page: 1  
  },  
  success: function(res) {  
    console.log(res)  
  }  
})
```

// 3.POST请求

```
wx.request({  
  url: 'https://httpbin.org/post',  
  data: {  
    name: 'why',  
    age: 18,  
    height: 1.88  
  },  
  method: 'post',  
  success: function(res) {  
    console.log(res)  
  }  
})
```




网络请求 – 请求封装

- 目前我们采用的网络请求是非常古老的请求方式, 我们将它封装成Promise的方式:

```
network.js  X
1  import {
2    |   baseUrl,
3    |   timeout
4  } from './config.js'
5
6  function request(options) {
7    |   return new Promise((resolve, reject) => {
8    |     |   wx.request({
9    |     |     |   url: baseUrl + options.url,
10    |     |     |   timeout: timeout,
11    |     |     |   data: options.data,
12    |     |     |   success: function(res) {
13    |     |     |     |   resolve(res.data)
14    |     |     |   },
15    |     |     |   fail: reject
16    |     |   })
17    |   })
18  }
19
20  export default request;
```



展示弹窗

- 小程序中展示弹窗有四种方式: showToast、showModal、showLoading、showActionSheet

```
<button size='mini' bindtap='onToastClick'>showToast</button>
<button size='mini' bindtap='onModalClick'>showModal</button>
<button size='mini' bindtap='onLoadingClick'>showLoading</button>
<button size='mini' bindtap='onActionSheetClick'>showActionSheet</button>
```

```
onToastClick() {
  wx.showToast({
    title: '你好啊',
    duration: 3000,
    icon: 'loading',
    mask: true,
    success: function() {
      console.log('展示成功')
    },
    fail: function() {
      console.log('展示失败')
    },
    complete: function() {
      console.log('展示完成')
    }
  })
},
```

```
onModalClick() {
  wx.showModal({
    title: '我是标题',
    content: '我是内容,哈哈',
    showCancel: true,
    cancelText: '返回',
    cancelColor: '#ff8800',
    success: function(res) {
      if (res.confirm) {
        console.log('用户点击了确定')
      }
      if (res.cancel) {
        console.log('用户点击了取消')
      }
    }
  })
},
```

```
onLoadingClick() {
  wx.showLoading({
    title: '加载中',
    mask: true
  })

  setTimeout(() => {
    wx.hideLoading()
  }, 2000)
},
```

```
onActionSheetClick() {
  wx.showActionSheet({
    itemList: ['相册', '拍照'],
    itemColor: '#ff0000',
    success: function(res) {
      console.log(res)
    }
  })
},
```




页面分享

- 分享是小程序扩散的一种重要方式，小程序中有两种分享方式：
 - 点击右上角的**菜单**按钮，之后点击**转发**
 - 点击某一个按钮，直接转发
- 当我们转发给好友一个小程序时，通常小程序中会显示一些信息：
 - 如何决定这些信息的展示呢？通过 **onShareAppMessage**

自定义转发内容

字段	说明	默认值	最低版本
title	转发标题	当前小程序名称	
path	转发路径	当前页面 path，必须是以 / 开头的完整路径	
imageUrl	自定义图片路径，可以是本地文件路径、代码包文件路径或者网络图片路径。支持PNG及JPG。显示图片长宽比是5:4。	使用默认截图	1.5.0

蘑菇街女装

夏装2019新款V领花边灯笼袖
洋气蕾丝衬衫女百搭超仙...



小程序

蘑菇街女装

蘑菇街女装



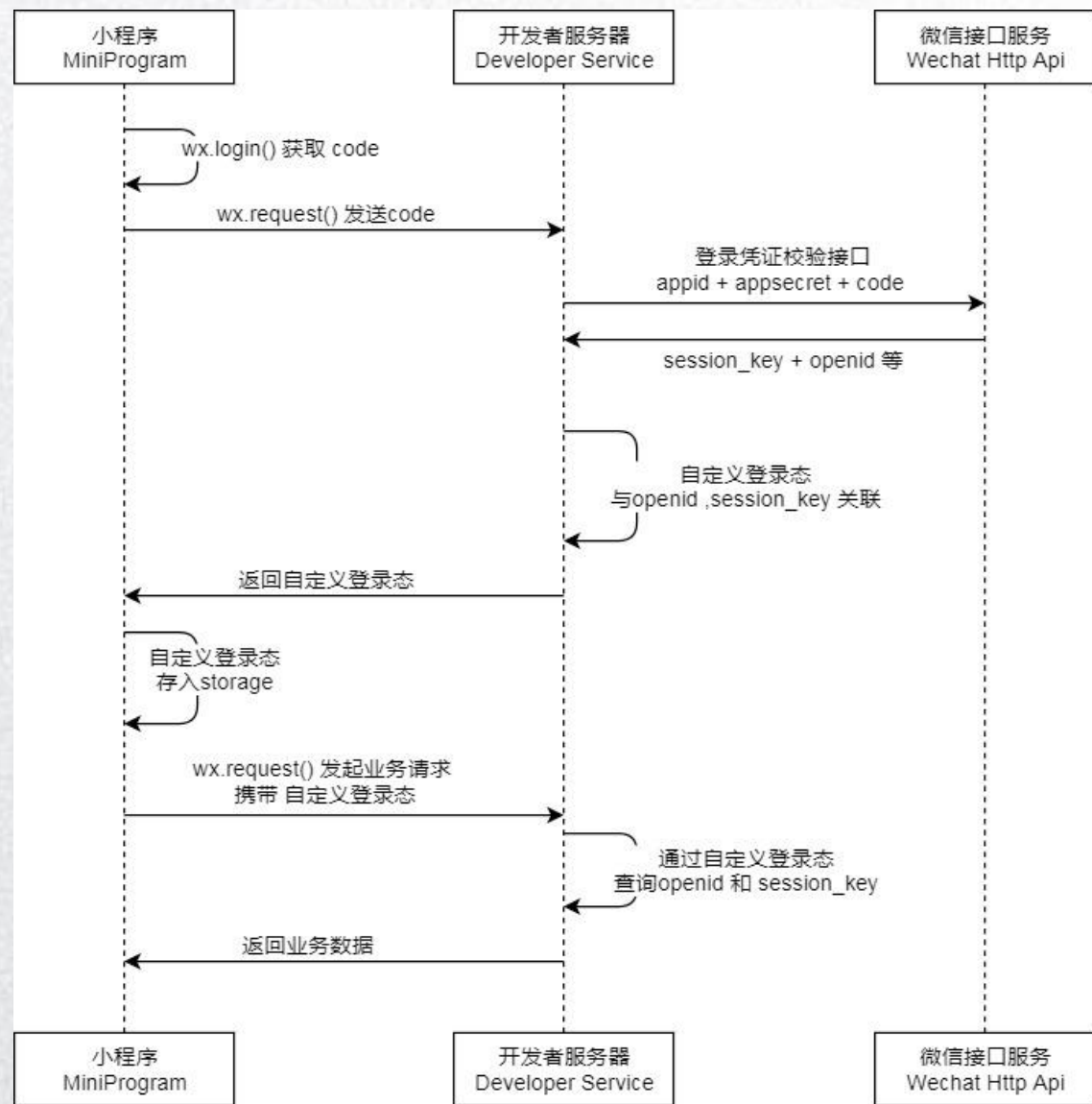
小程序



小程序的登录流程

■ 我们先来查看一下官方给出的小程序登录流程图：

- ❑ 1.调用wx.login获取code
- ❑ 2.调用wx.request发送code到我们自己的服务器（我们自己的服务器会返回一个登录态的标识，比如token）
- ❑ 3.将登录态的标识token进行存储，以便下次使用
- ❑ 4.请求需要登录态标识的接口时，携带token





小程序登录演练

请求token接口

接口地址: /login

请求方式: post

参数列表: code

返回值: token

验证token接口

接口: /auth

请求: post

参数: header, token: 本地保存的token

返回值:

错误码:

1001 没有传入token

1002 传入错误的 token

1003 token过期

```
// 获取code
wx.login({
  success: (res) => {
    // 1.获取code
    const code = res.code;

    // 2.向自己的服务器发送请求
    wx.request({
      url: 'http://123.207.32.32:3000/login',
      method: 'post',
      data: {
        code
      },
      success: (res) => {
        console.log(res)
        // 1.取出token
        const token = res.data.token

        // 2.存储到全局变量中
        this.globalData.token = token

        // 3.存储到storage中
        wx.setStorage({
          key: TOKEN,
          data: token,
          success: function(res) {
            console.log(res)
          }
        })
      }
    })
  }
})
```

```
onLaunch: function () {
  // 1.从缓存中取出token
  const token = wx.getStorageSync(TOKEN)

  // 2.判断是否取出token
  if (token && token.length) {
    this.check_token(token)
  } else {
    console.log('进行了登录')
    this.login()
  }
},

check_token(token) {
  console.log(token)
  wx.request({
    url: 'http://123.207.32.32:3000/auth',
    method: 'post',
    header: {
      token
    },
    success: (res) => {
      console.log(res)
    }
  })
}
```




界面跳转

- 界面的跳转有两种方式：**通过navigator组件** 和 **通过wx的API跳转**
- navigator组件主要就是用于界面的跳转的：

属性	类型	默认值	必填	说明
target	string	self	否	在哪个目标上发生跳转，默认当前小程序
url	string		否	当前小程序内的跳转链接
open-type	string	navigate	否	跳转方式
delta	number	1	否	当 open-type 为 'navigateBack' 时有效，表示回退的层数



open-type的取值

■ open-type有如下取值：

open-type 的合法值

值	说明	最低版本
navigate	对应 <code>wx.navigateTo</code> 或 <code>wx.navigateToMiniProgram</code> 的功能	
redirect	对应 <code>wx.redirectTo</code> 的功能	
switchTab	对应 <code>wx.switchTab</code> 的功能	
reLaunch	对应 <code>wx.reLaunch</code> 的功能	1.1.0
navigateBack	对应 <code>wx.navigateBack</code> 的功能	1.1.0
exit	退出小程序, <code>target="miniProgram"</code> 时生效	2.1.0

- **redirect**: 关闭当前页面，跳转到应用内的某个页面。但是不允许跳转到 tabbar 页面，并且不能返回。（不是一个压栈）
- **switchTab**: 跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面。（需要在tabBar中定义的）
- **reLaunch**: 关闭所有的页面，打开应用中某个页面。（直接展示某个页面，并且可以跳转到tabBar页面）



导航返回

■ 导航返回有两个属性来起作用：

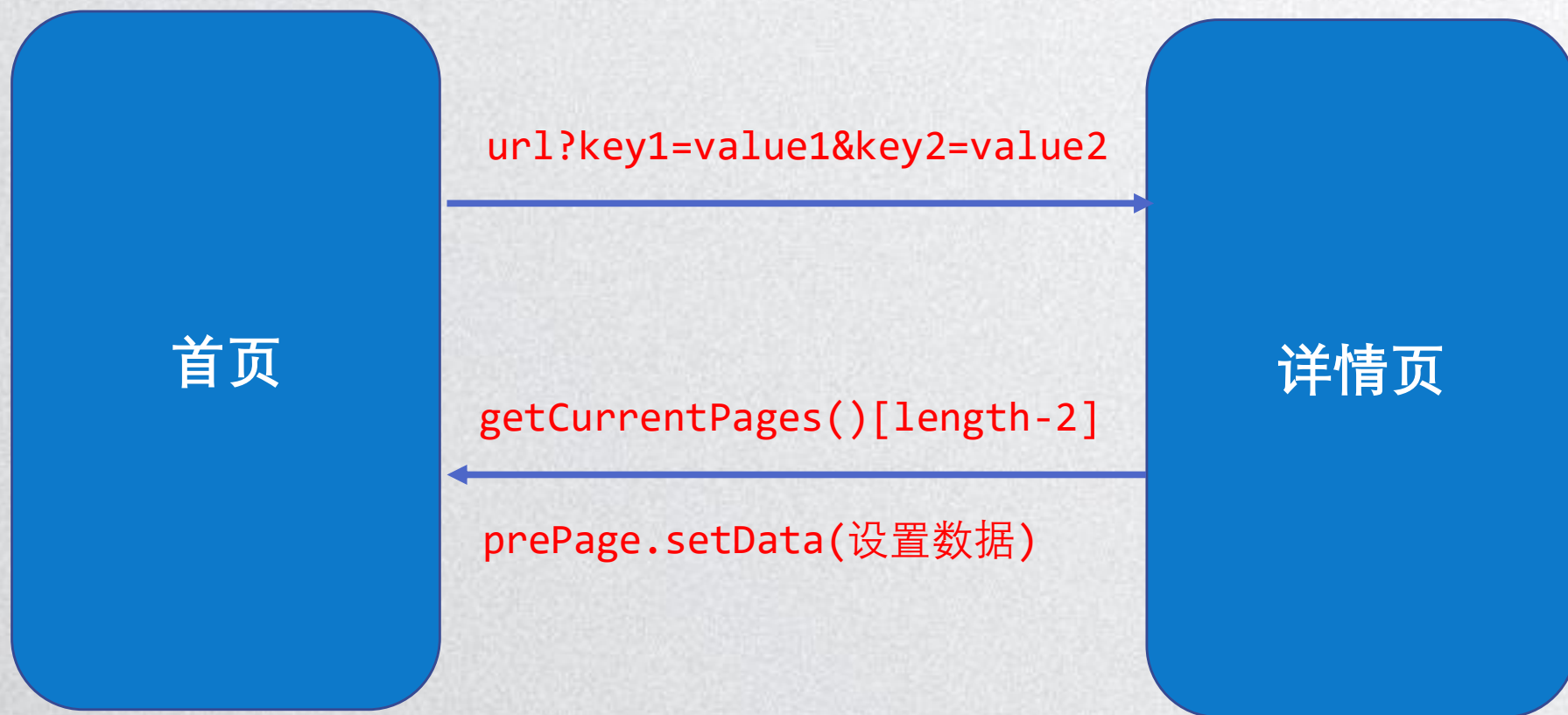
- open-type: navigateBack (表示该navigator组件用于返回)
- delta: 返回的层级 (指定返回的层级, open-type必须是navigateBack才生效)

```
<!-- 返回上一级 -->  
<navigator open-type='navigateBack'>返回上一级</navigator>
```




数据传递 – 传递方式分析

- 如何在界面跳转过程中我们需要相互传递一些数据，应该如何完成呢？
 - 首页 -> 详情页：使用URL中的query字段
 - 详情页 -> 首页：在详情页内部拿到首页的页面对象，直接修改数据





数据传递 – 传递过程

■ 首页 -> 详情页:

- 通过修改URL传递参数

■ 详情页 -> 首页:

- 返回时携带数据有两个问题需要考虑

■ 问题一: 在什么地方修改数据

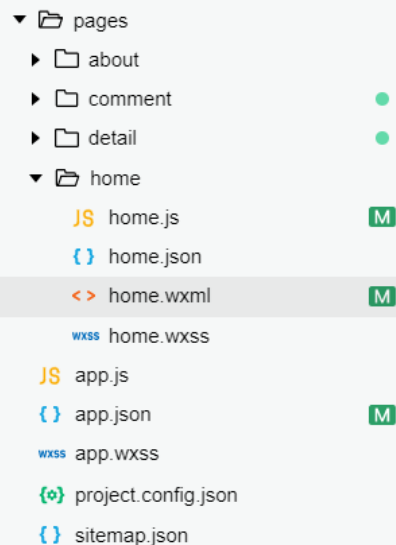
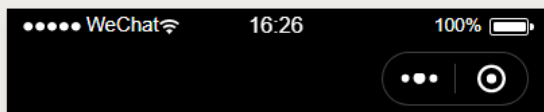
- - 如果你是监听按钮或者navigator的点击来返回时, 可以通过bindtap来映射到某个函数, 在函数中完成.
- - 但是这种方式不能监听左上角返回按钮的点击.
- - 所以我们选择在onUnload中修改数据

■ 问题二: 如何修改数据

- - 小程序并没有提供直接修改数据的方法.
- - 但是可以通过getCurrentPages来获取所有的页面, 然后使用页面对象的setData({})函数来修改



代码演练



```
1 <!--pages/home/home.wxml-->|
2 <text>Hello World</text>
3 <view class='title'>你好,小程序</view>
4 <view>{{message}}</view>
5
6 <!-- 1.简单的跳转 -->
7 <navigator url='../detail/detail'>跳转到详情页</navigator>
8
9 <!-- 2.跳转到tab页面 -->
10 <navigator url='../about/about' open-type='switchTab'>跳转到关于</navigator>
11
12 <!-- 3.重新启动,加载某一个页面 -->
13 <navigator url='../comment/comment' open-type='reLaunch'>重新显示评论</navigator>
14
15 <!-- 4.跳转到详情页, 携带数据 -->
16 <navigator url='../detail/detail?name=why&age=18'>跳到详情页,并且携带数据</navigator>
17
18 <!-- 5.通过button进行页面跳转 -->
19 <button size='mini' bindtap='onBtnClick'>跳转到详情页</button>
20
```




代码的跳转和返回

- 很多情况下，我们并不喜欢使用navigator组件来进行跳转：
 - 可能我们希望用户点击了某个button或者view时，对该button或者view进行监听
 - 之后，通过相关的代码逻辑实现跳转
- 对此，微信也提供了对应的API接口：
 - `wx.navigateTo(url[,])`
 - `wx.navigateBack([delta])`

```
onBtnClick() {  
  wx.navigateTo({  
    url: '../detail/detail?name=kobe&age=30',  
  })  
}
```

```
onBack() {  
  wx.navigateBack({  
    delta: 1  
  })  
}
```