



数据存储与访问

数据存储是开发应用程序时需要解决的最基本问题，数据必须以某种方式保存，并且能够有效、方便地使用和更新处理。微信小程序开发框架提供的本地存储主要包含本地缓存存储访问机制和文件系统存储访问机制。本章将结合具体的案例介绍本地缓存存储访问机制和文件系统存储访问机制的原理和使用方法。

本章学习目标

- 了解小程序本地存储与访问数据的原理；
- 掌握数据缓存 API 和文件 API 的使用方法；
- 掌握 `wx.chooseImage()`、`wx.getImageInfo()`、`wx.saveImageToPhotosAlbum()`、`wx.previewImage()` 等图片管理 API 的使用方法和应用场景；
- 掌握 `wx.getLocation()`、`wx.openLocation()`、`wx.chooseLocation()` 等位置信息 API 的使用方法和应用场景。



5.1 概述



进行各类应用开发时，涉及的数据存储访问方式主要有两类：本地存储和网络存储。本地存储方式多用于离线应用开发，网络存储方式多用于需要及时更新数据的重要项目事务，比如在线售票、天气预报等应用中的实时数据需要通过网络传输到数据处理中心存储和处理。

微信小程开发框架提供的本地存储主要包含以下两种存储访问机制。



5.1

5.1.1 本地缓存存储访问机制

每个微信小程序都有自己的本地缓存，同一个微信用户、同一个小程序拥有不超过 10MB 的本地缓存空间。同一台设备上不同用户的同一个小程序都分别拥有自己的本地缓存空间，该缓存空间不仅互相隔离，而且不能互相访问。

如果设备存储空间不足，微信小程序会清空最近最久未使用的本地缓存。因此开发者进行微信小程序开发时，通常只能使用本地缓存机制存储数据量不大的一些非关键信息，以防设备存储空间不足，导致数据丢失、用户更换设备导致数据不能迁移等问题。

5.1.2 文件系统存储访问机制

微信小程序的文件系统管理操作的文件分为代码包文件和本地文件两类。

(1) 代码包文件。

代码包文件指的是微信小程序项目开发时向项目目录中添加的文件，用户对该类文件只



具有读权限。对于需要动态替换或占用存储空间较大的文件，尽量不要使用这种存储访问机制。如 image 组件引用的本地图片文件，需要保存到项目的指定目录中才能由 src 属性引用。

代码包文件的访问路径必须从项目根目录开始写文件路径，不支持相对路径的写法。如：/a/b/c、a/b/c 都是合法的，./a/b/c、../a/b/c 则不合法。

(2) 本地文件。

本地文件是指通过微信小程序开发框架提供的接口 API 在本地创建或 HTTPS 请求网络下载并存储到本地的文件。具体包括：

① 本地临时文件：它是临时产生，又可能随时会被回收的文件，文件大小不受限制；本地临时文件的访问路径为 {{协议名}}://tmp/文件名。

② 本地缓存文件：微信小程序通过调用接口，把本地临时文件缓存后产生的文件，不能自定义文件夹名称和文件名称，它会随小程序的删除而删除。普通小程序的本地缓存文件大小与本地用户文件大小合计最多 10MB，游戏类小程序的本地缓存文件大小与本地用户文件大小合计最多 50MB；本地缓存文件的访问路径为 {{协议名}}://store/文件名。

③ 本地用户文件：微信小程序通过调用接口，把本地临时文件缓存后产生的文件，允许自定义文件夹名称和文件名称，文件大小与本地缓存文件一样；本地用户文件的访问路径为 {{协议名}}://usr/文件名。

其中，协议名在 iOS/Android 客户端为 wxfile，而在开发者工具上为 http，开发者不用关注这个差异。

当微信小程序被用户添加到设备后，会有一块独立的文件存储区域，并与用户维度隔离。即同一个设备，每个微信用户不能访问到其他登录用户的文件，同一个用户不同 appId 之间的文件也不能互相访问。每个微信用户对本地临时文件和本地缓存文件只具有读权限，而对本地用户文件既有读权限，又有写权限。

微信小程开发框架提供的网络存储方式有 HTTPS 网络请求和云数据库存储访问机制两种。关于网络数据存储访问机制，将在第 8 章介绍。

5.2 随手拍的设计与实现



现在越来越多的人喜欢用手机、平板电脑等智能终端设备拍照，但拍完照片后，时间长了容易忘记拍的地点、时间或具体内容等。“随手拍”小程序可以让使用者随时随地拍下精彩瞬间，并且可以同时记录下拍摄的地点、时间，也可以给照片添加详细的说明信息。本节以开发设计“随手拍”的案例介绍小程序实现本地缓存存储访问机制的原理和使用方法。



5.2.1.1

5.2.1 预备知识

1 图片 API

小程序的图片应用是一个常用的技术，比如调用设备的摄像头拍摄图片，选择喜欢的图片发朋友圈、保存图片等。微信小程序开发框架提供了专门对图片进行处理的 API。下面以实现图 5.1 所示小程序为例，介绍选择/拍摄图片、预览图片、获取图片信息、压缩图片、保存图片到系统相册等 API 的使用方法。

单击图 5.1 所示界面的“拍照”按钮，调用设备的摄像头拍照，完成后照片显示在界面下部的图片显示区域；单击“选择图片”按钮，弹出“相册选择”对话框，在对话框中选择



图 5.1 图片 API 应用（1）

需要的图片，其显示在界面下部的图片显示区域；单击“查看图片信息”按钮，在控制台打印图片的长和宽的像素值及图片文件的路径；单击“保存”按钮，将图片保存到系统相册中；单击 image 组件，在新页面进行图片预览。页面结构文件代码如下：

```
1 <!--pages/photo/photo.wxml-->
2 <button bindtap='getPic'>拍照</button>
3 <button bindtap='choosePic'>选择图片</button>
4 <button bindtap='showPicInfo'>查看图片信息</button>
5 <button bindtap='savePic'>保存</button>
6 <image bindtap='viewPic' src='{{picPath}}' /></image>
```

（1）wx.chooseImage（Object object）：从本地相册选择图片或使用相机拍照。object 参数及功能说明如表 5-1 所示。

表 5-1 object 参数及功能说明

属 性	类 型	功 能
count	Number	最多可选择的图片张数，默认值为 9
sizeType	Array<string>	所选的图片类型，默认值为['original', 'compressed']
sourceType	Array<string>	所选的图片来源，默认值为['album', 'camera']
success	Function	接口调用成功的回调函数
fail	Function	接口调用失败的回调函数
complete	Function	接口调用结束的回调函数

original 表示原图，compressed 表示压缩图，album 表示相册，camera 表示相机。success(res) 回调函数的 res.tempFilePaths 返回图片的本地临时文件路径列表、res.tempFiles 返回图片的本地临时文件列表、res.tempFiles[index].path 返回本地临时文件列表中下标为 index 的图片文件路径、res.tempFiles[index].size 返回图片的本地临时文件列表中下标为 index 的图片文件大小（单位：Byte）。因此，实现拍照功能的 getPic()函数代码如下：

```
1 getPic() {
```



```
2   var that = this
3   wx.chooseImage({
4     count: 4,
5     sizeType: ['original', 'compressed'],
6     sourceType: ['camera'],
7     success: function (res) {
8       var imgs = res.tempFilePaths
9       var tem = res.tempFilePaths[0]
10      that.setData({
11        picPath: tem,
12        imgUrls: imgs
13      })
14    }
15  })
16 },
```

其中第 4 行代码表示可选择的图片数最多为 4 张；第 5 行代码表示选择的图片为原图或压缩图类型；第 6 行代码表示可选择的图片来源于相机；第 7~14 行代码表示调用成功后，将图片文件路径列表下标为 0 的元素（即当前照片的文件路径）及图片文件路径列表通过 setData() 函数更新。

wx.chooseImage() 函数也可以从相册选择图片，所以实现选择图片功能的 choosePic() 函数代码与 getPic() 函数类似，只要将上述代码第 6 行的 camera 修改为 album 即可。当然，如果选择图片时既可以从“相册”也可以从“相机”，就可以将上述第 6 行代码修改为 sourceType: ['album', 'camera']，这样在单击“选择图片”按钮后，界面底部会弹出图 5.2 所示的选择按钮。



图 5.2 图片 API 应用（2）

(2) wx.getImageInfo (Object object): 获取图片信息。object 参数及功能说明如表 5-2 所示。



表 5-2 object 参数及功能说明

属 性	类 型	功 能
src	String	图片的路径
success	Function	接口调用成功的回调函数
fail	Function	接口调用失败的回调函数
complete	Function	接口调用结束的回调函数

src 表示要获得图片信息的图片文件路径, 该路径可以是相对路径、临时文件路径、存储文件路径、网络图片路径。网络图片需配置 download 域名后才能生效, 其内容会在后面网络应用开发章节介绍。success(res)回调函数的 res.width 和 res.height 返回图片的原始宽度和高度(单位: px), res.path 返回图片的本地路径, res.orientation 返回拍照时的设备方向, res.type 返回图片格式。因此, 实现显示图片信息功能的 showPicInfo ()函数代码如下:

```
1 showPicInfo: function () {
2   var that = this
3   wx.getImageInfo({
4     src: that.data.picPath,
5     success: function (res) {
6       console.log('图片宽: ', res.width, "图片高: ", res.height, "图片路径: ",
7       res.path)
8     }
9   })
10 }
```

其中第 4 行代码指定图片路径的 src 为 picPath (由 data 定义); 第 6 行代码表示在控制台容器输出图片的宽度、高度和图片的路径。

(3) wx.saveImageToPhotosAlbum(Object object): 保存图片到系统相册。object 参数及功能说明如表 5-3 所示。

表 5-3 object 参数及功能说明

属 性	类 型	功 能
filePath	String	图片文件路径
success	Function	接口调用成功的回调函数
fail	Function	接口调用失败的回调函数
complete	Function	接口调用结束的回调函数

filePath 表示要保存到系统相册的图片文件路径, 该路径可以是临时文件路径或永久文件路径, 但不支持网络图片路径; success(res)和 fail(res)回调函数的 res.errMsg 返回保存图片的返回信息。因此, 实现保存图片功能的 savePic ()函数代码如下:

```
1 savePic: function () {
2   var that = this
3   wx.saveImageToPhotosAlbum({
4     filePath: that.data.picPath, //要保存图片的路径
5     success: function (res) {
6       console.log(res.errMsg) //输出保存成功的回调信息
7       wx.showToast({
8         title: '保存成功!',
```



```
9      })
10     },
11     fail: function (res) {
12       console.log(res.errMsg)    //输出保存失败的回调信息
13       wx.showToast({
14         title: '保存失败! ',
15       })
16     }
17   })
18 },
```

(4) wx.previewImage(Object object): 在新页面中全屏预览图片。预览的过程中，用户可以进行保存图片、发送给朋友等操作。object 参数及功能说明如表 5-4 所示。

表 5-4 object 参数及功能说明

属 性	类 型	功 能
urls	Array<string>	要预览的图片链接列表
current	String	当前显示的图片链接，默认值为 urls 中的第一项
success	Function	接口调用成功的回调函数
fail	Function	接口调用失败的回调函数
complete	Function	接口调用结束的回调函数

urls 表示要全屏预览图片的链接地址列表，该列表中必须存放网络地址。因此，实现在新页面预览图片功能的 viewPic ()函数代码如下：

```
1  viewPic: function () {
2    var that = this
3    wx.previewImage({
4      current: that.data.imgUrls[0], //当前显示图片的 http 链接
5      urls: that.data.imgUrls        //需要预览的图片 http 链接列表
6    })
7  },
```

2 位置 API

位置应用也是微信小程序的一种常用功能，微信小程序框架提供了使用微信内置地图查看位置、获得当前位置及选择位置的 API。

(1) wx.getLocation(Object object): 获取当前地理位置、速度等信息。object 参数及功能说明如表 5-5 所示。



5.2.1.2

表 5-5 object 参数及功能说明

属 性	类 型	功 能
type	String	坐标类型，wgs84 类型返回 gps 坐标，gcj02 类型返回国测局坐标，默认值为 wgs84
altitude	Boolean	是否返回海拔高度，true 返回高度信息，由于获取高度需要较高精确度，因此会减慢接口返回速度，默认值为 false
success	Function	接口调用成功的回调函数
fail	Function	接口调用失败的回调函数
complete	Function	接口调用结束的回调函数



空间测量都需要一个特定的坐标系作为基准，微信小程序使用的坐标系统有 wgs84 和 gcj02 两种标准：wgs84 的全称是 World Geodetic System 1984，它是美国国防局为 GPS（全球定位系统）在 1984 年建立的一种地心坐标系；gcj02 的全称是国家测量局 02 标准，它是中国国家测量局定制的信息系统坐标系。目前，微信 Web 开发者工具仅支持 gcj02 坐标，并且 gcj02 国测局坐标可用于 wx.openLocation() 函数的坐标。其代码使用格式如下：

```
1 wx.getLocation({
2   type: 'gcj02', //设置坐标类型
3   altitude: true, //设置返回高度信息
4   success(res) {
5     const latitude = res.latitude //纬度
6     const longitude = res.longitude //经度
7     const speed = res.speed //速度
8     const accuracy = res.accuracy //位置精确度
9     const altitude = res.altitude //位置高度
10  }
11 })
```

wx.getLocation() 函数中 success(res) 回调函数的 res.latitude 和 res.longitude 返回当前位置的纬度（范围为 -90~90，负数表示南纬）和经度（范围为 -180~180，负数表示西经），res.speed 返回速度（单位为 m/s），res.accuracy 返回精确度，res.altitude（返回当前位置高度，单位为 m，开发中必须使用上述第 3 行代码）。

（2）wx.openLocation(Object object)：根据纬度、经度，在微信内置地图查看位置。object 参数及功能说明如表 5-6 所示。

表 5-6 object 参数及功能说明

属 性	类 型	功 能
latitude	Number	设置要查看位置的纬度
longitude	Number	设置要查看位置的经度
scale	Number	设置地图缩放比例，范围为 5~18，默认值为 18
name	String	设置查看位置的位置名
address	String	设置查看位置的地址详细说明
success	Function	接口调用成功的回调函数
fail	Function	接口调用失败的回调函数
complete	Function	接口调用结束的回调函数

wx.openLocation() 函数的使用代码格式如下：

```
1 wx.openLocation({
2   latitude: latitude, //设置要查看位置的纬度
3   longitude: longitude, //设置要查看位置的经度
4   scale: 6, //设置地图缩放比例
5   name: '我的位置', //设置要查看位置的名称
6   address: '图书馆大楼 9 楼', //设置要查看位置的详细地址信息
7   success: function(res) {
8     console.log(res.errMsg) //输出函数调用结果信息
9   }
10 })
```

上述代码第 2 行用于指定要查看位置的纬度值；第 3 行用于指定要查看位置的经度值；第 5~6 行用于指定在地图底部显示的“名称”和“地址信息”，运行效果如图 5.3 所示。



图 5.3 位置 API 应用（1）

（3）wx.chooseLocation(Object object)：打开微信内嵌地图选择位置。object 参数及功能说明如表 5-7 所示。

表 5-7 object 参数及功能说明

属 性	类 型	功 能
success	Function	接口调用成功的回调函数
fail	Function	接口调用失败的回调函数
complete	Function	接口调用结束的回调函数

wx.chooseLocation()函数的使用代码格式如下：

```
1 wx.chooseLocation({
2   success: function(res) {
3     var name = res.name           //返回位置名称
4     var address = res.address     //返回位置详细信息
5     var latitude = res.latitude  //返回位置纬度
6     var longitude= res.longitude //返回位置经度
7     console.log( name, address,latitude,longitude)
8   },
9 })
```

运行上述代码后，根据当前位置列出附近地址信息，供用户选择。用户选择地址后，success(res)回调函数的 res.name 返回选中位置的名称，res.address 返回选中位置的详细信息，



res.latitude 和 res.longitude 分别返回选中位置的纬度和经度。运行效果如图 5.4 所示。



图 5.4 位置 API 应用（2）

3 数据缓存 API

移动端应用程序经常需要访问一些业务数据，这些数据通常数量较大、访问频率较高。如果都使用网络访问机制存储和访问这些数据，一方面会浪费网络带宽，另一方面也会降低小程序的运行效率。HTML5 开始提供了一种在客户端存储数据的新方法——`localStorage`，它突破了传统的 `cookie` 存储数据的 4KB 限制。使用它进行数据存储，就相当于针对前端页面的小型数据库。



5.2.1.3

微信小程序开发框架也提供了 `localStorage` 的数据存储访问机制，该机制将数据存储在本机缓存的指定 `key` 中。数据存储生命周期跟小程序本身一致，即除用户主动删除或超过一定时间被自动清理，数据都一直可以使用。单个 `key` 允许存储的最大数据长度为 1MB，所有数据存储上限为 10MB。

（1）`wx.setStorage (Object object)`：向本地缓存中异步存储数据。`object` 参数及功能说明如表 5-8 所示。

表 5-8 object 参数及功能说明

属 性	类 型	功 能
key	String	设置本地缓存中用于存储数据的键（key）
data	Any	设置本地缓存中存储的数据值（value）
success	Function	接口调用成功的回调函数
fail	Function	接口调用失败的回调函数
complete	Function	接口调用结束的回调函数

`wx.setStorage()`函数的使用代码格式如下：

```
1 wx.setStorage({
```



```
2      key: 'age',    //保存的 key 名为 age
3      data: 20,      //保存的数据为 20
4      success(res) { //保存成功的回调函数
5          wx.showToast({
6              title: '保存成功',
7          }),
8      },
9  }
```

上述代码表示在本地缓存中存放 1 个 key 为 age，值为 20 的数值；如果存放成功，调用 wx.showToast()函数给出提示信息。代码运行后调试窗口显示结果如图 5.5 所示。

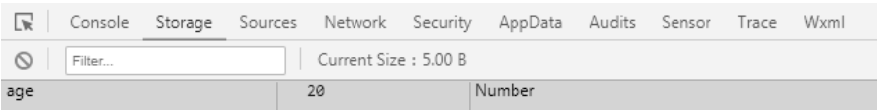


图 5.5 本地缓存数据

(2) wx.setStorageSync(string key, any data): 向本地缓存中同步存储数据。该函数的使用代码格式如下：

```
1 wx.setStorageSync('name', 'kate')
```

上述代码表示在本地缓存中存放 1 个 key 为 name，值为 kate 的字符串。

(3) wx.getStorage(Object object): 从本地缓存中异步读取数据。object 参数及功能说明如表 5-9 所示。

表 5-9 object 参数及功能说明

属 性	类 型	功 能
key	String	设置从本地缓存中读取键(key)对应的数据
success	Function	接口调用成功的回调函数
fail	Function	接口调用失败的回调函数
complete	Function	接口调用结束的回调函数

success(res)回调函数的 res.data 返回 key 对应的数据。wx.getStorage()函数的使用代码格式如下：

```
1 wx.getStorage({
2     key: 'age',
3     success: function(res) {
4         console.log(res.data)
5     },
6     fail:function(res){
7         console.log(res.errMsg)
8     }
9 })
```

上述代码第 2 行指定要读本地缓存中 key 为 age 的值；第 3~5 行代码表示读取成功后，在控制台窗口输出 age 的值；第 6~8 行代码表示读取失败后，在控制台窗口输出错误信息。

(4) wx.getStorageSync(string key, any data): 从本地缓存中同步读取数据。该函数的使用代码格式如下：