

一、button组件

事件绑定: `bindtap="tapName"`

如: `<button bindtap="addTodo">添加</button>`

、 `<view bindtap="tapName">Click me!</view>`

open-type 有效值: **getUserInfo**获取用户信息、**openSetting**授权设置、**share**转发

二、登录用户信息存取实现

1、通过按钮登录来获取用户登录信息（如：昵称、所在城市、头像、性别等）

`<!-- 否则 渲染登录按钮 -->`

`<button wx:else open-type = "getUserInfo" bindgetuserinfo="OnGetUserInfo">`

// 获取用户信息的回调函数

`OnGetUserInfo(e){`

`// console.log('获取到的用户信息',e.detail)`

`// 云函数获取用户ID`

`wx.cloud.callFunction({`

`name:'login',`

`data:{`

`},`

`success:(res)=>{`

`// 登录成功后获取用户登录信息`

`let userInfo = e.detail.userInfo`

`// 给userInfo这个数据设置一个新字段openid`

`userInfo.openid = res.result.OPENID`

`// 重新设置数据`

```
this.setData({  
  userInfo  
})  
// 将登录信息存储到数据中  
wx.setStorageSync('userInfo', userInfo);  
// console.log('登录成功的新数据',userInfo)  
}  
})  
},
```

2、通过云函数login来获取用户登录的私密信息（用户ID、应用ID、支付相关ID）

```
// 云函数入口文件  
const cloud = require('wx-server-sdk')  
  
cloud.init()  
  
// 云函数入口函数  
exports.main = async (event, context) => {  
  // cloud.getWXContext() 获取用户的私密信息  
  const { OPENID } = cloud.getWXContext()  
  return {  
    event,  
    OPENID  
  }  
  // openid: wxContext.OPENID,  
  // appid: wxContext.APPID,  
  // unionid: wxContext.UNIONID,  
}  
}
```

3、组装登录信息和私密信息到一起

```
wx.cloud.callFunction({
  name:'login',
  data:{
  },
  success:(res)=>{
    // 登录成功后获取用户登录信息
    let userInfo = e.detail.userInfo
    // 给userInfo这个数据设置一个新字段openid
    userInfo.openid = res.result.OPENID
    // 重新设置数据
    this.setData({
      userInfo
    })
    // 将登录信息存储到数据中
    wx.setStorageSync('userInfo', userInfo);
    // console.log('登录成功的新数据',userInfo)
  }
})
```

4、将登录成功后，组装好的登录信息存储在Storage缓存中

```
// 将登录信息存储到数据中
wx.setStorageSync('userInfo', userInfo);
// console.log('登录成功的新数据',userInfo)
}
```

5、在全局app.js页面加载的函数中获取Storage缓存的数据并设置变量

同步获取： wx.getStorageSync(string key) 异步获取： wx.getStorage(Object
object)

同步存: `wx.setStorageSync(string key, any data)` 异步

存: `wx.setStorage(Object object)`

// 在全局获取登录信息并存入全局变量中

```
const userInfo = wx.getStorageSync('userInfo');
```

```
if(userInfo && userInfo.openid){
```

```
// console.log('本地存储',userInfo)
```

```
// this.setData({
```

```
// userInfo
```

```
// })
```

```
this.globalData.userInfo = userInfo;
```

```
}
```

6、在全局js中的生命周期函数中获取Storage缓存并入全局变量中

```
App({
```

```
// 生命周期函数
```

```
onLaunch: function () {
```

```
wx.cloud.init({
```

```
traceUser:true //这个用户追踪可以不用
```

```
})
```

```
// 在全局获取登录信息并存入全局变量中
```

```
const userInfo = wx.getStorageSync('userInfo');
```

```
if(userInfo && userInfo.openid){
```

```
// console.log('本地存储',userInfo)
```

```
// this.setData({
```

```
// userInfo
```

```
// })
```

```
this.globalData.userInfo = userInfo;
```

```
}
```

7、在需要用到的页面中获取全局变量

// 将全局信息设置给当前页面的数据变量

```
this.setData({  
  userInfo:app.globalData.userInfo  
})
```

8、条件渲染显示页面

<!-- 如果 userInfo.openid 有值 渲染 用户昵称 登录成功-->

<text wx:if="{{userInfo.openid}}">恭喜您！ {{userInfo.nickName}}登录成功</text>

<!-- 否则 渲染登录按钮 -->

<button wx:else open-type = "getUserInfo" bindgetuserinfo="OnGetUserInfo">

登录

</button>

<image src="{{userInfo.avatarUrl}}" />

三、后端数据库的使用

1、进入云开发控制台----->数据库----->添加集合。定义新的数据库名

2、获取操作数据库的实例

// 操作数据库 获得实例

```
const db = wx.cloud.database()
```

3、操作数据表

// 通过实例操作指定的数据表

```
db.collection('goodlist').add({
```

```
  data:{
```

```
    name:this.data.knames,
```

```
    price:this.data.prices,
```

```

count:this.data.counts,

solded:5,

icomPath:this.data.fileP,

catg:['前端','VUEjs','源码']

},

success(res){

wx.showToast({

title:"添加成功了"

})

}

})

```

4、上传图片 在小程序端可以分别调

用 **wx.cloud.uploadFile** 和 **wx.cloud.downloadFile** 完成上传和下载云文件操作

// 选择图片

```

wx.chooseImage({

count: 1,

sizeType: ['original', 'compressed'],

sourceType: ['album', 'camera'],

success:img=>{

// tempFilePath可以作为img标签的src属性显示图片

const tempFilePaths = img.tempFilePaths

console.log('图片地址: ',tempFilePaths)

// 获取图片上传过来的地址 filePath cloudPath 固定字段不可改

const filePath = img.tempFilePaths[0]

// 截取成数组，数组中包含文件扩展名

const temp = filePath.split('.')

// 重定义图片名

const cloudPath = this.data.knames+'.'+temp[temp.length-1]

```

// 上传图片

```
wx.cloud.uploadFile({  
  //云存储文件名  
  cloudPath,  
  filePath,  
  success:res=>{  
    console.log('返回结果ID',res.fileID)  
    this.setData({  
      fileP:res.fileID // 设置图片地址  
    })  
    wx.showToast({  
      title:"上传成功了"  
    })  
  }  
})
```

5、显示图片<image> mode 有 13 种模式，其中 4 种是缩放模式，9 种是裁剪模式

数据库的增查改删操作：

查：

```
// 获取数据库中所有数据  
getGoodList(){  
  const db = wx.cloud.database()  
  db.collection('goodlist').where({  
    _openid:app.globalData.userInfo.openid  
  }).get({
```

```
success:res=>{  
  this.setData({  
    goodlist:res.data  
  })  
  // 修改全局数据 在别处要用全局数据  
  app.globalData.goodlists = this.data.goodlist  
  // console.log('获取到的数据',this.data.goodlist)  
  // console.log('获取到的全局数据',app.globalData.goodlists)  
}  
})  
},
```

增:

```
// 操作数据库 获得实例  
const db = wx.cloud.database()  
// 通过实例操作指定的数据表  
db.collection('goodlist').add({  
  data:{  
    name:this.data.knames,  
    price:this.data.prices,  
    count:this.data.counts,  
    solded:5,  
    icomPath:this.data.fileP,  
    catg:['前端','VUEjs','源码']  
  },  
  success(res){  
    wx.showToast({  
      title:"添加成功了"
```



```
})  
}  
})
```

改:

// 获取要修改的数据

```
const goodExa = this.data.goodlist[0]
```

// 判断是否有数据可修改

```
if(goodExa){
```

// 开始修改数据

```
const db = wx.cloud.database()
```

```
db.collection('goodlist').doc(goodExa._id)
```

```
.update({
```

```
data:{
```

```
name:this.data.xg
```

```
},
```

```
success:res=>{
```

```
wx.showToast({
```

```
title:"修改成功了"
```

```
})
```

```
}
```

```
})
```

```
}else{
```

```
wx.showToast({
```

```
title:"没有数据"
```

```
})
```

```
}
```

//完成修改后清空输入框数据

this.remXg()