

全栈小程序开发

1. 课前准备

2. 回顾

3. 课堂主题

4. 课堂目标

5. 知识点

云函数引入

获取用户信息

获取openid

持久化存储openid

查看用户

用户状态查询

添加云数据库

添加集合

新增数据

上传文件

读取数据

更新数据

下拉刷新

触底加载

6. 扩展点

7. 总结

8. 作业 && 答疑

9. 下节课预告

1. 课前准备

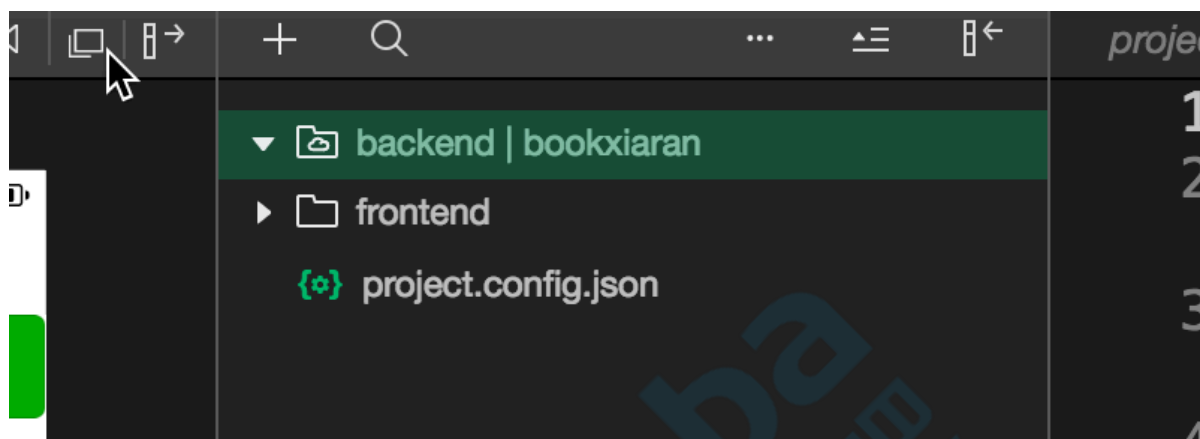
1. 云开发文档 <https://developers.weixin.qq.com/miniprogram/dev/wxcloud/basis/getting-started.html>

filePath: http://tmp/wxeb3481f0cde7006f.o6zAJs_c_gB5x-OteY9wCi0We7l4.FiYbOkvF9mdAe88a7728f76b1200115ae17c03da924f.png

cloudPath my-image123.png

Nodejs

- 鉴权
- 持久化 - mysql mongodb
- 资源文件
- 日志和统计



2. 回顾

1. 小程序开发流程
2. 小程序tabbar配置
3. 小程序常用组件
4. 小程序常用api
5. 云开发

3. 课堂主题

1. 云开发实战
2. 如何使用云存储
3. 如何使用云数据库 新增数据
4. 如何查询云数据库的数据
5. 开发登录云函数
6. 开发商品列表

4. 课堂目标

1. 开发自己的云函数
2. 开发自己的全栈小程序

5. 知识点

云函数引入

增加后台

```
// 除project.config.json文件外所有文件全部放到 frontend中
```

```
//project.config.json
{
  "miniprogramRoot": "mini-fe/",
  "cloudfunctionRoot": "mini-back/",
}
```

```
//backend/sum/index.js
```

```
// 云函数入口文件
```

```
const cloud = require('wx-server-sdk')
```

```
cloud.init()
```

```
// 云函数入口函数
```

```
exports.main = (event, context) => {
  console.log('console.log',event)
  console.log(context)
  return {
    sum: event.a + event.b
  }
}
```

```
// app.js
```

```
onLaunch: function () {
  console.log('onLaunch...')
  // 这里添加
  wx.cloud.init()
},
```

```
wx.cloud.callFunction({
  // 云函数名称
  name: 'add',
  // 传给云函数的参数
  data: {
    a: 1,
    b: 2,
  },
  success(res) {
    console.log(res.result.sum) // 3
  },
  fail: console.error
})
```

```
wx.cloud.callFunction({
  // 云函数名称
  name: 'add',
  // 传给云函数的参数
  data: {
    a: 1,
    b: 2,
  },
})
.then(res => {
  console.log(res.result) // 3
})
.catch(console.error)
```

获取用户信息

getUserInfo必须要用button来获取，得到用户授权后，可以通过wx.getUserInfo获取，不会有弹窗提示

```
// pages/logs/logs.wxml
<image src="{{userInfo.avatarUrl}}" />
<view wx:if="{{userInfo.nickName}}">{{userInfo.nickName}}</view>
<button wx:else open-type="getUserInfo" bindgetuserinfo="onGetUserInfo">
  登录
</button>
```

```
// pages/logs/logs.js
Page({
  data: {
    userInfo: {}
  },
  onGetUserInfo(e) {
    console.log('coGetUserInfo', e)
    if (e.detail.userInfo) {
      this.setData({
        avatarUrl: e.detail.userInfo.avatarUrl,
        userInfo: e.detail.userInfo
      })
    }
  }
})
```

获取openid

用户的openid，是当前用户在小程序里的唯一标识，使用云开发，可以很方便的实现获取openid 也就是登录的功能，借助wx-server-sdk

新建云函数

```
/server-end/login/login.js
const cloud = require('wx-server-sdk')
cloud.init()
exports.main = (event, context) => {
  const wxContext = cloud.getWXContext()
  return {
    event,
    openid: wxContext.OPENID,
  }
}
```

使用

```
// pages/logs/logs.js
onGetUserInfo(e){
  // ....
  // 添加
  wx.cloud.callFunction({
    name: 'login',
    data: {},
    success: res => {
      console.log(res)
      console.log('[云函数] [login] user openid: ', res.result.openid)

    },
    fail: err => {
      console.error('[云函数] [login] 调用失败', err)
      wx.navigateTo({
        url: '../deployFunctions/deployFunctions',
      })
    }
  })
  // ....
}
```

持久化存储openid

```
wx.cloud.callFunction({
  name: 'login',
  data: {},
  success: (res)=>{
    let userInfo = e.detail.userInfo
    userInfo.openid = res.result.openid
    this.setData({
      userInfo
    })

    wx.setStorageSync('userInfo', userInfo)
    console.log('登录成功', res)
  }
})
```

查看用户



用户状态查询

```
onLoad: function(){
  const userInfo = wx.getStorageSync('userInfo')
  console.log('userInfo', userInfo)
  if(userInfo && userInfo.openid){
    this.setData({userInfo})
  }
},
```

添加云数据库

<https://developers.weixin.qq.com/miniprogram/dev/wxcloud/guide/database/getting-started.html>

一个类似mongodb的数据库 和传统的mysql关系型数据库的区别

关系型	文档型
数据库 database	数据库 database
表 table	集合 collection
行 row	记录 record / doc
列 column	字段 field

添加集合

添加商品集合 goods 云开发后台操作

新增数据

文档 <https://developers.weixin.qq.com/miniprogram/dev/wxcloud/guide/database/add.html>

```
const db = wx.cloud.database()
db.collection('goods').add({
  data: {
    name: 'web全栈',
    count: 1
  },
  success: res => {
    // 在返回结果中会包含新创建的记录的 _id
    console.log(res)
    wx.showToast({
      title: '新增记录成功',
    })
    console.log('[数据库] [新增记录] 成功, 记录 _id: ', res._id)
  },
  fail: err => {
    wx.showToast({
      icon: 'none',
      title: '新增记录失败'
    })
    console.error('[数据库] [新增记录] 失败: ', err)
```



```
}  
})
```

上传文件

使用云开发的云存储能力

wx.chooseImage 选择图片

wx.cloud.uploadFile上传文件

```
<button bindtap="doUpload">上传头图</button>  
<image src="{{uploadUrl}}"></image>
```

```
doUpload() {  
  // 选择图片  
  wx.chooseImage({  
    count: 1,  
    sizeType: ['compressed'],  
    sourceType: ['album', 'camera'],  
    success: res => {  
      wx.showLoading({  
        title: '上传中',  
      })  
      const filePath = res.tempFilePaths[0]  
      // 上传图片  
      const cloudPath = 'my-image123' + filePath.match(/\.([^.]+)$/)[0]  
      wx.cloud.uploadFile({  
        cloudPath,  
        filePath,  
        success: res => {  
          console.log('[上传文件] 成功: ', res)  
          this.setData({  
            uploadUrl: res.fileID  
          })  
        },  
        complete: () => {  
          wx.hideLoading()  
        }  
      })  
    },  
  })  
},  
})  
},
```

读取数据

在记录和集合上都有提供 `get` 方法用于获取单个记录或集合中多个记录的数据。

```
db.collection('todos').where({
  _id: 'id'
})
.get({
  success(res) {
    // res.data 是包含以上定义的两条记录的数组
    console.log(res.data)
  }
})
```

更新数据

使用update来更新数据，更新方式多样化，可以直接修改新的值，也支持递增这些函数操作

更新指令	说明
set	设置字段为指定值
remove	删除字段
inc	原子自增字段值
mul	原子自乘字段值
push	如字段值为数组，往数组尾部增加指定值
pop	如字段值为数组，从数组尾部删除一个元素
shift	如字段值为数组，从数组头部删除一个元素
unshift	如字段值为数组，往数组头部增加指定值

列表页

```
// frontend/pages/list/list.js
Page({

  /**
   * 页面的初始数据
   */
  data: {
    goods: []
  },
  onLoad(options) {
    this.getList()
  },
  getList() {
    const db = wx.cloud.database();
    db.collection('goods').get({
      success: res => {
        console.log('res', res);
        this.setData({ goods: res.data });
      }
    });
  }
})
```

```
<view wx:for="{{goods}}" wx:key="{{item.id}}">
  <view>{{index}}:{{item.name}}</view>
  <image src="{{item.image}}"></image>
</view>
```

比如每次访问, count+1

```
const _ = db.command
db.collection('goods').doc('goodid').update({
  data: {
    //新的值
    name: 'xx',
    // 表示指示数据库将字段自增 10
    count: db.command.inc(1)
  },
  success(res) {
    console.log(res.data)
  }
})
```

下拉刷新

<https://developers.weixin.qq.com/miniprogram/dev/framework/config.html#%E9%A1%B5%E9%9D%A2%E9%85%8D%E7%BD%AE>

需要配置 enablePullDownRefresh 配合onPullDownRefresh 生命周期

```
onPullDownRefresh() {
  // Do something when pull down.
},
```

触底加载

需要配置 onReachBottomDistance 配置onReachBottom

```
onReachBottom() {
  // Do something when page reach bottom.
},
```

6. 扩展点

1. 权限校验
2. 云开发函数去获取网络接口数据

7. 总结

1. 回顾知识点
2. 提示学习方法
3. 提示本次课重点和必会知识

全栈小程序开发

1. 课前准备
2. 回顾
3. 课堂主题
4. 课堂目标
5. 知识点

云函数引入
获取用户信息
获取openid
持久化存储openid
查看用户
用户状态查询
添加云数据库
添加集合
新增数据
上传文件
读取数据
更新数据

下拉刷新
触底加载

6. 扩展点
7. 总结
8. 作业 && 答疑
9. 下节课预告

8. 作业 && 答疑

9. 下节课预告

1. taro 多端框架

filePath: http://tmp/wxeb3481f0cde7006f.o6zAJs_c_gB5x-OteY9wCi0We7l4.ugZlw0dRO46Oda278e53732c9ecf25bcb12064e4c656.png

