

## 全栈小程序开发

### 1. 课前准备

### 2. 课堂主题

### 3. 课堂目标

### 4. 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

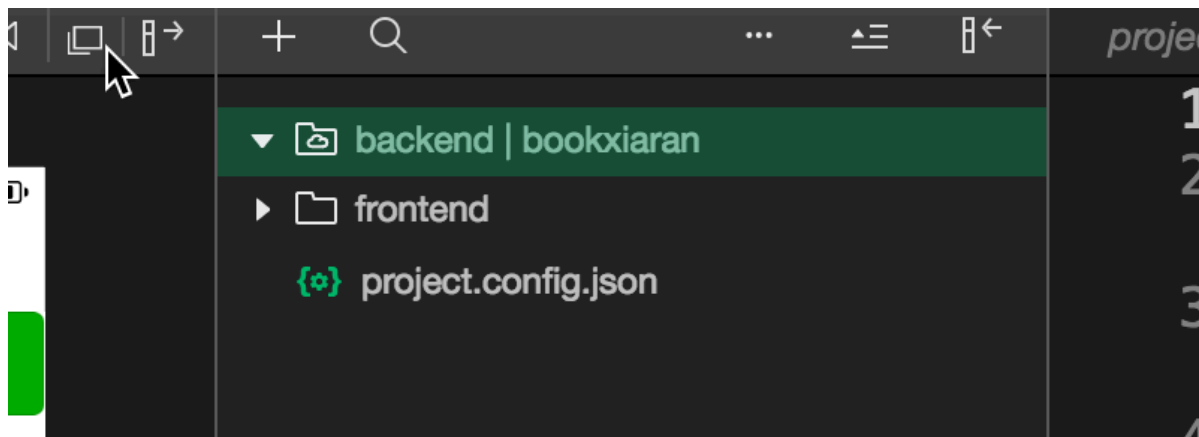
### 5. 扩展点

## 1. 课前准备

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

### Nodejs

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



## 2. 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

## 3. 课堂目标

1. 开发自己的云函数
2. 开发自己的全栈小程序

## 4. 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

## 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

云开发控制台

当前环境: kaikeba

概览 用户管理 数据库 存储管理 云函数 统计分析

用户OpenID/用户昵称

头像	昵称	性别	城市	注册时间	最近一次进入时间	操作
	前端	男	-	2019-02-19 11:37:23	2019-02-22 19:03:14	复制openid

共 1 项

每页显示行 20

## 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>

## 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

## 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.  
},
```

## 5. 扩展点

---

1. 权限校验
2. 云开发函数去获取网络接口数据

