



云开发公开课

30 分钟搭建个人相册小程序



目录

一、实验概述	2
二、准备工作	2
三、实验架构	2
四、任务一：创建小程序·云开发环境.....	3
任务目标: 创建小程序·云开发环境，用于后面储存相册的用户信息和照片。	3
1、开发者工具创建项目	3
2、开通云开发环境	3
3、创建数据库	5
五、任务二：搭建个人相册	6
任务目标: 创建个人相册，实现照片的上传和存储。	6
1、打开项目目录下的 app.js 文件，修改初始化云函数配置中的 env 参数为 上一任务中创建的环境 ID，并保存。	6
2、实现了用户登录的能力，我们需要接着来实现照片上传的能力.....	8

一、实验概述

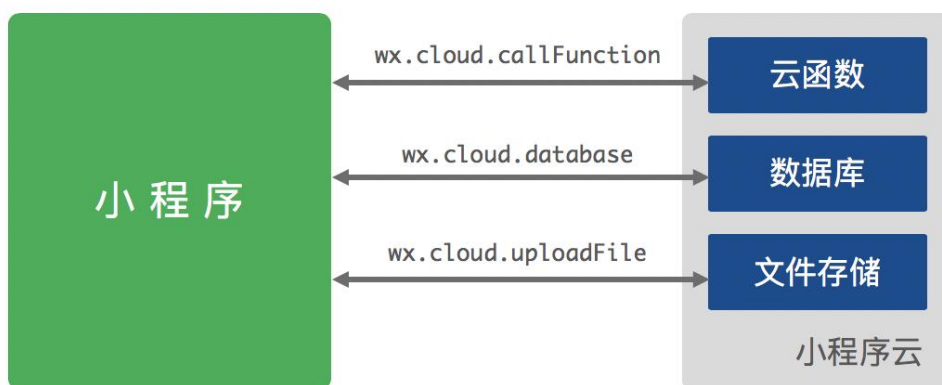
图片存储，是所有应用开发里最常见的场景之一。

本次腾讯云 TechWork，将通过实战“个人相册小程序”开发，教你如何借助小程序云开发能力，提升功能开发效率，提升数据隐私保护能力。

二、准备工作

1. 已申请小程序云开发公测资格的微信小程序账号
2. 公测版本的微信开发者工具
3. 下载个人相册小程序 Demo 代码

三、实验架构



四、任务一：创建小程序·云开发环境

任务目标：创建小程序·云开发环境，用于后面储存相册的用户信息和照片。

1、开发者工具创建项目

打开微信开发者工具，创建一个新的小程序项目，项目目录选择个人相册 Demo 的目录，AppID 填写已经申请公测资格的小程序对应的 AppID。



The screenshot shows the '小程序项目管理' (Mini Program Project Management) dialog box in the WeChat Developer Tools. It features a purple icon with a white 'S' shape. The title is '小程序项目' (Mini Program Project) with a subtitle '编辑、调试小程序' (Edit, debug Mini Program). There are three input fields: '项目目录' (Project Directory) with a dropdown menu showing '选择项目目录' (Select Project Directory); 'AppID' with a text box containing '已申请小程序·云开发公测资格的 AppID' (AppID for Mini Program CloudBase public test qualification) and a link '若无 AppID 可 注册' (If no AppID, you can register) or '或使用测试号: 小程序 / 小游戏' (Or use test number: Mini Program / Mini Game); and '项目名称' (Project Name) with a text box containing '任意填写项目名称' (Enter any project name). A green '确定' (Confirm) button is at the bottom right.

2、开通云开发环境

- 1) 点击开发者工具上的【云开发】按钮

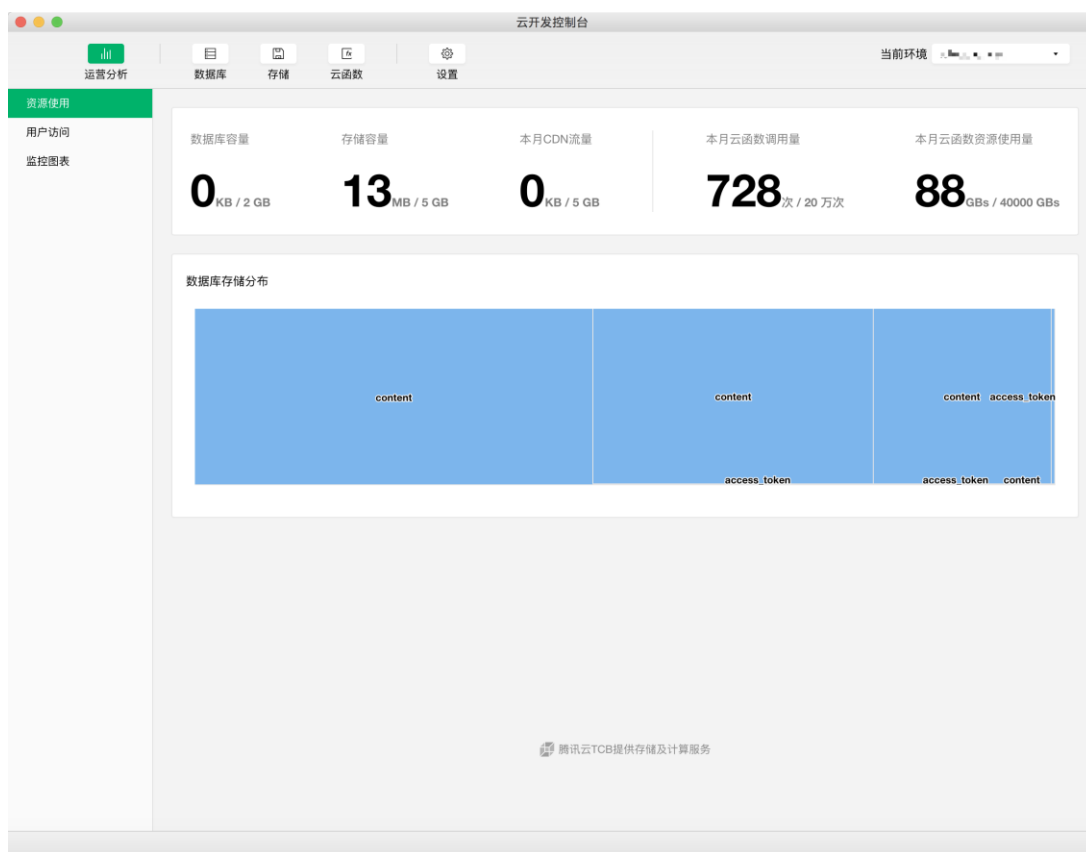


2) 点击【同意】



3) 填写环境名称和环境 ID，点击【确定】创建环境，即可进入云开发控制台。





3、创建数据库

相册小程序会使用到云开发提供的数据库能力，数据库使用的是 MongoDB，需要优先创建一个集合，方便之后使用。

- 1) 打开云开发控制台，点击菜单栏中的【数据库】，然后点击左侧边栏的【添加集合】按钮



2) 输入集合名称 “user”，然后点击确定即可创建集合。

创建集合

集合名称

user

支持英文字母大小写、数字、- 和 _

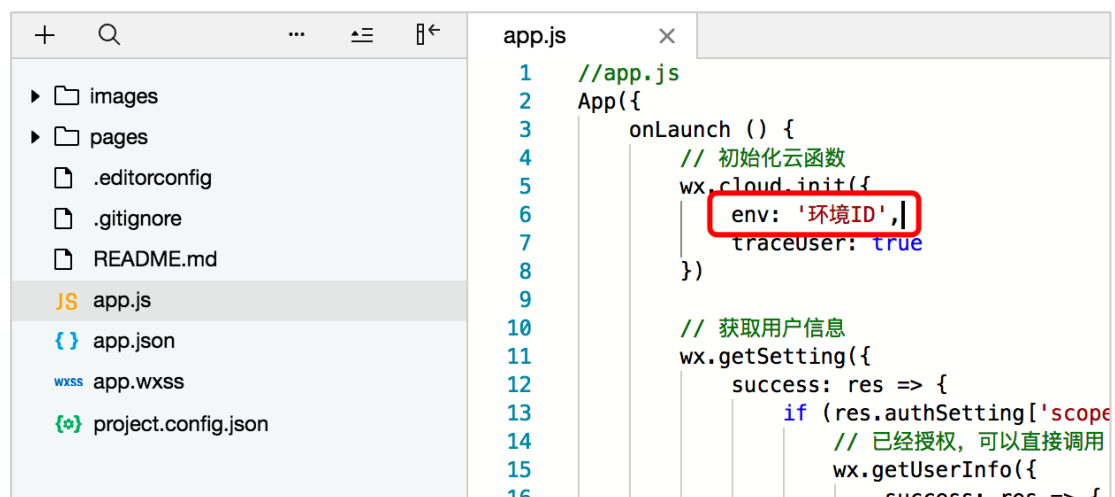
取消

确定

五、任务二：搭建个人相册

任务目标: 创建个人相册，实现照片的上传和存储。

1、打开项目目录下的 `app.js` 文件，修改初始化云函数配置中的 `env` 参数为上一任务中创建的环境 ID，并保存。



打开 `pages/user/user.js` 文件，里面是用户登录所相关的 js 逻辑，我们需要在文件中的 `addUser` 函数里添加保存用户信息到数据库的逻辑，代码如下：

```

// 获取数据库实例

const db = wx.cloud.database({})

// 插入用户信息

let result = await db.collection('user').add({

  data: {

    nickName: user.nickName,

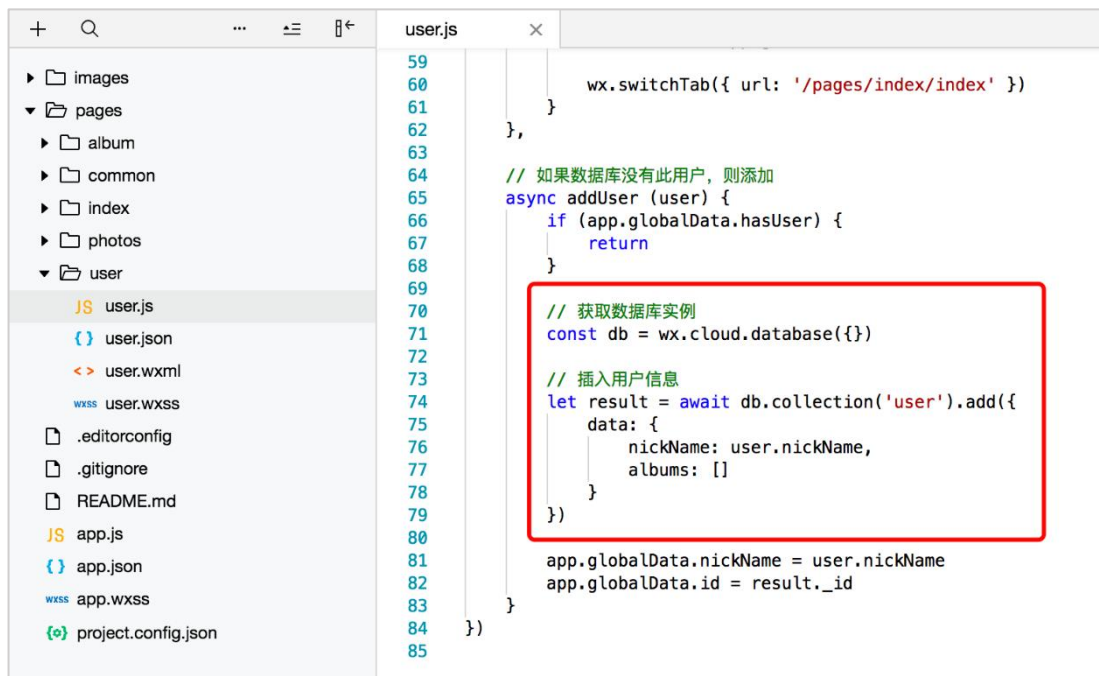
    albums: []

  }

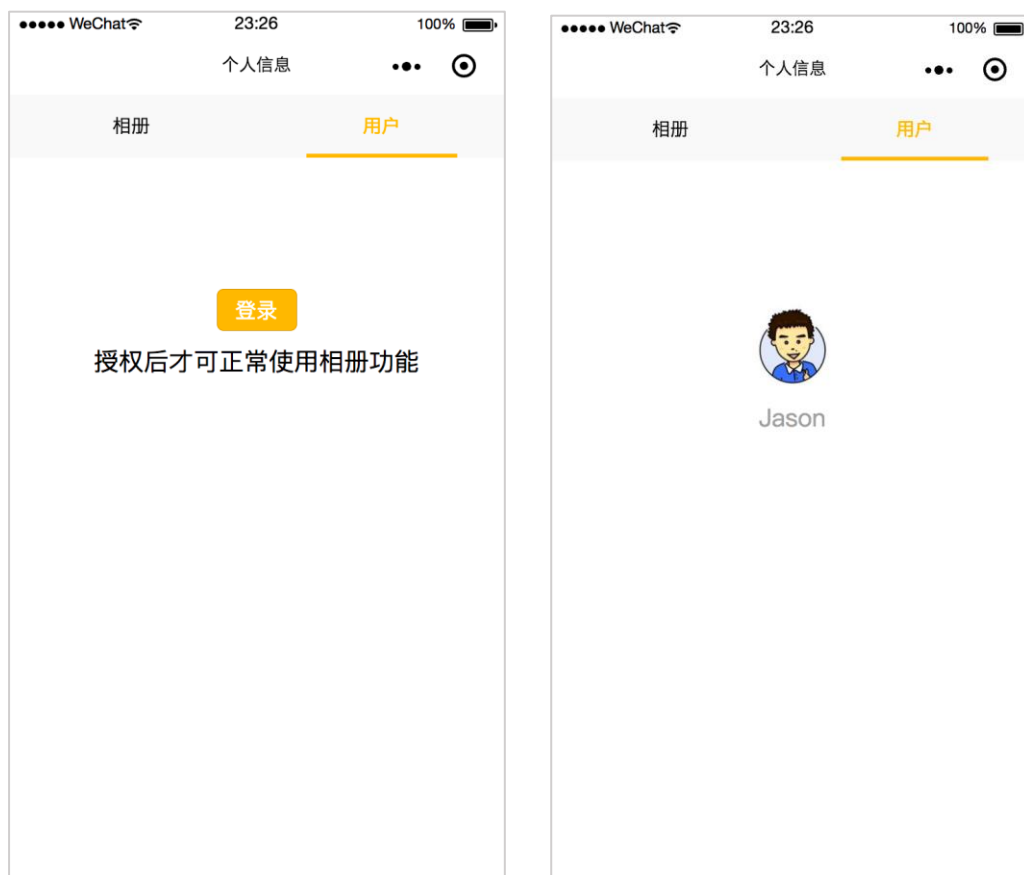
})

```

复制粘贴在文件的 70 行：



保存文件后，就实现了用户登录的能力。页面会自动刷新，点击页面上的登录按钮，即可在页面上看到自己的昵称和头像。

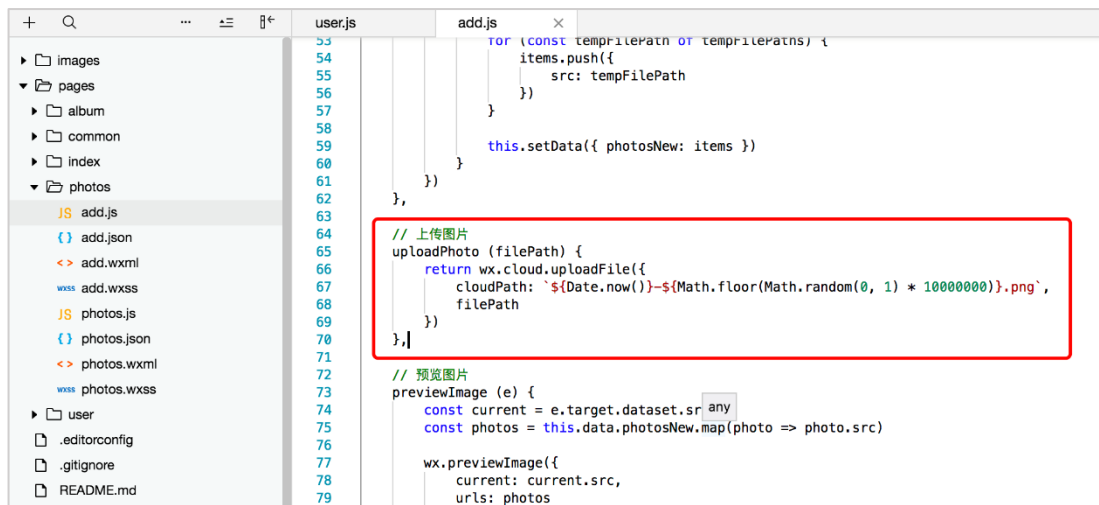


2、实现了用户登录的能力, 我们需要接着来实现照片上传的能力

照片选择和上传的相关代码在 `pages/photos/add.js` 中, 打开文件, 找到 `uploadPhoto` 的函数, 即可看到函数接收了一个 `filePath` 的参数, 他是用户选择照片时照片的本地临时路径, 我们需要使用云能力将图片上传到文件储存中, 代码如下:

```
// 调用 wx.cloud.uploadFile 上传文件
return wx.cloud.uploadFile({
  cloudPath: `${Date.now()}-${Math.floor(Math.random(0, 1) * 10000000)}.png`,
  filePath
})
```

粘贴代码在文件的 66 行：



保存文件后，上传的能力就完成了。文件上传后，和登录一样，我们需要将上传好的文件信息存储在数据库中，这个逻辑在 pages/photos/add.js 文件的 addPhotos 函数完成。相册的数据存储在用户信息中，函数已经帮我们完成了对用户信息的更新，我们只需要完成用户信息的更新即可，代码如下：

```

// 写入集合

db.collection('user').doc(app.globalData.id).update({

  data: {

    albums: db.command.set(app.globalData.allData.albums)

  }

}).then(result => {

  console.log('写入成功', result)

  wx.navigateBack()

})

```

粘贴代码到文件的 102 行：



```
100  })
101  },
102  // 添加图片信息到数据库
103  addPhotos (photos, comment) {
104    // 初始化数据库
105    const db = wx.cloud.database({})
106    const oldPhotos = app.globalData.allData.albums[this.data.albumIndex].photos
107    const albumPhotos = photos.map(photo => ({
108      fileId: photo.fileID,
109      comments: comment
110    }))
111    app.globalData.allData.albums[this.data.albumIndex].photos = [...oldPhotos, ...albumPhotos]
112
113    // 写入集合
114    db.collection('user').doc(app.globalData.id).update({
115      data: {
116        albums: db.command.set(app.globalData.allData.albums)
117      }
118    }).then(result => {
119      console.log('写入成功', result)
120      wx.navigateBack()
121    })
122  }
123 }
```

保存文件后，页面会自动刷新。至此，我们已经完成了一个简单的小程序的搭建，并将最核心的能力使用小程序云开发完成，快使用微信开发者工具或者扫描开发二维码用手机体验吧！



扫码关注腾讯云「云开发」

我们一起探讨大前端 serverless 的未来