



云开发实践

智能名片识别小程序



目录

1 项目概述	2
2 任务目标	2
3 准备工作	2
4 实战架构	2
5 实战任务	3
5.1 修复名片图片上传功能	3
5.2 修复名片识别功能	4
5.3 修复名片删除功能	5
6 完成标准	6
7 结果页面	6

1 项目概述

还为收集到太多名片而无处存放感到苦恼？基于人工智能的名片识别小程序能帮到你。

利用云开发的各種能力，轻松实现名片内容识别，名片信息管理等功能。

通过本项目的简略开发，带你领略借助小程序·云开发能力，提升功能开发效率。

2 任务目标

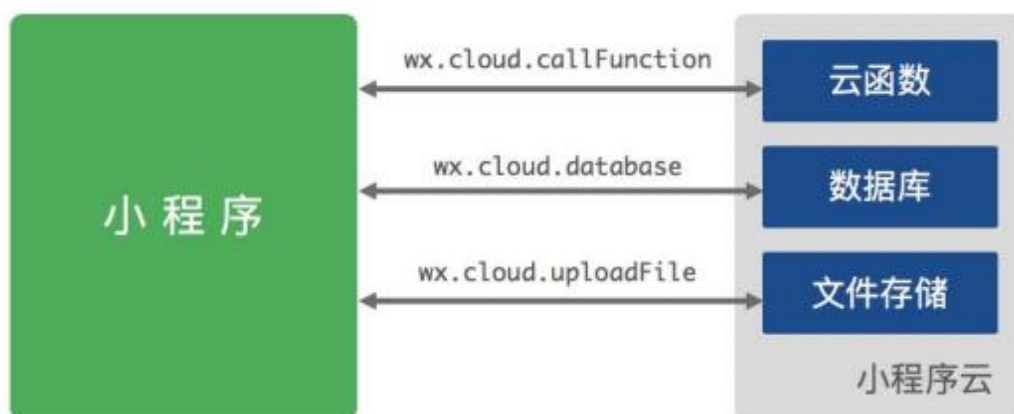
根据项目初始框架，完善代码；使得名片内容识别功能能够正常使用，完成智能名片识别小程序的开发，并预览至手机进行名片小程序的体验。

3 准备工作

打开微信小程序开发者工具，选择【智能名片识别】项目，等待项目加载完成。

如果有任何异常，请联系周边工作人员寻求帮助。

4 实战架构



5 实战任务

5.1 修复名片图片上传功能

1、打开 `miniprogram/pages/index/index.js` 文件，定位到【代码实战位置 A-1】，此位置所在的函数作用为名片图片上传；在使用 `wx.chooseImage` 选择图片后，得到的回调参数中有 `tempFilePaths`，包含着选择图片的临时地址。

```
uploadFile() {
  let that = this;
  if(app.globalData.id==null){
    wx.showToast({
      title: '初始化未完成，请稍后再试',
      icon: 'none'
    });
    return;
  }
  wx.chooseImage({
    success: function (dRes) {
      wx.showLoading({
        title: '上传文件',
      })
      let cloudPath = `${app.globalData.id}/${Date.now()}-${Math.floor(Math.random(0, 1)*1000)}.png`;

      /*【开始】=====【代码实战位置A-1】=====【开始】*/

      /*【结束】=====【代码实战位置A-1】=====【结束】*/

    },
    fail: console.error
  })
},
},
```

2、将红框中的代码填写到【代码实战位置 A-1】开始和结束之间，使整个函数流程完整。空缺的代码是云开发在小程序端提供的云存储能力，通过调用 `wx.cloud.uploadFile`，即可实现文件存储云端的过程，在存储之后通过 `then` 和 `catch` 来继续进行流程逻辑。

```
/*【开始】=====【代码实战位置A-1】=====【开始】*/
wx.cloud.uploadFile({
  cloudPath,
  filePath: dRes.tempFilePaths[0],
}).then(res => {
  if (res.statusCode < 300) {
    that.setData({
      fileID: res.fileID,
    }) => { that.getTempFileURL()
  }
}).catch(err => {
  wx.hideLoading()
  wx.showToast({ title: '上传失败', icon: 'none'})
})
/*【结束】=====【代码实战位置A-1】=====【结束】*/
```

5.2 修复名片识别功能

1.我们在上传名片图片后，需要利用 AI 识别名片接口进行名片识别。在这里，我们利用云函数来完成名片识别的功能。打开 `miniprogram/pages/index/index.js` 文件，定位到【代码观看位置 A-2】，此为小程序发起云函数请求操作，这将调用名字为 `parseNameCard` 的云函数执行识别任务，并将结果返回。

```
/*【开始】=====【代码观看位置A-2】=====【开始】*/
wx.cloud.callFunction({
  name: 'parseNameCard',
  data: {
    url: that.data.coverImage
  }
}).then(res => {
  console.log(res)
  if (res.result.code !== 0) {
```

2.打开 `cloudfunctions/parseNameCard/index.js`，定位到【代码实战位置 B-1】，将红框中的代码填写到【代码实战位置 B-1】开始和结束之间，使整个函数完整。

```
exports.main = async (event) => {
  /*【开始】=====【代码实战位置B-1】=====【开始】*/
  const res = await imgClient.ocrBizCard({
    data: {
      url_list: [event.url],
    },
  });
  return JSON.parse(res.body).result_list[0];
  /*【结束】=====【代码实战位置B-1】=====【结束】*/
}
```

*****右键文件列表 `parseNameCard` 文件夹，选择【上传并部署：云端安装依赖】上传。**

3.我们在这里使用了 `image-node-sdk` 包，是腾讯云封装的图像服务，在此包使用前，需要在腾讯云开通相关服务，并在腾讯云账户中获得密钥信息，才可使用。关于此代码包的具体信息可以扫码查看。



5.3 修复名片删除功能

1.我们在点开名片详情后，可以操作使当前名片删除，我们依然利用云函数来完成名片单一删除功能。打开 `miniprogram/pages/detail/detail.js` 文件，定位到【代码观看位置 D-1】，此为小程序发起云函数请求操作，这将调用名字为 `remove` 的云函数执行删除名片图片任务，我们传入图片的链接地址作为删除的名片信息标识。

2.打开 `cloudfunctions/remove/index.js`，定位到【代码实战位置 C-1】，将红框中的代码填写到【代码实战位置 C-1】开始和结束之间，使整个逻辑完整。(注释字不用填入，作为解释)

```
try {
  /*【开始】=====【代码实战位置C-1】=====【开始】*/

  //利用数据库where查找，数据集中openid为用户所属的文档，然后使用update进行更新操作
  const userdata = await db.collection('list').where({
    openid: event.userInfo.openId
  }).update({
    data: {
      //_.pull为command能力，含义为将数组中src为event.img的元素删除
      list: _.pull({
        cover: event.img
      })
    }
  });
  //使用云存储能力，根据列表的fileid地址删除文件
  await cloud.deleteFile({
    fileList: [event.img]
  })

  /*【结束】=====【代码实战位置C-1】=====【结束】*/
}
```

*** 右键文件列表 `remove` 文件夹，选择【上传并部署：云端安装依赖】上传云函数。

3.remove 函数使用数据库操作符 `command.pull` 进行数组更新。即将名片数组中匹配 `cover` 为传入 `img` 的元素移除掉，做到了名片的删除。在代码里的下划线 (`_`) 为一个变量，在文件第 14 行进行了 `const _ = db.command` 赋值操作。

如果上面的步骤你都已经完成了，你已经完成了智能名片识别小程序的搭建，并将功能使用小程序·云开发完成，快使用微信开发者工具或者使用手机扫码预览体验吧！

6 完成标准

- 1、名片小程序的功能都可以正常使用。
- 2、名片识别数据使用云数据库能力存储在云端，并从云端拉取展示
- 3、名片图片借助云存储能力存储在云端，并从云端拉取展示
- 4、任何页面没有 Warning 报错

7 结果页面



上小程序，用云开发



关注腾讯云「云开发」
上小程序，用云开发