



云开发公开课

AI 智能名片识别小程序



目录

一、实验概述.....	2
二、准备工作.....	2
三、效果预览.....	3
四、知识点.....	3
五、任务一：创建小程序·云开发环境.....	4
六、任务二：上传名片	7
七、任务三：使用云函数识别名片	8
八、任务四：读取名片列表与详情	9
九、预览.....	11

一、实验概述

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。

二、准备工作

已经申请小程序，获取小程序AppID 在小程序管理后台中，[设置]→[开发设置] 下可以获取微信小程序AppID。



1. 腾讯云账号，获取腾讯云的 APPID, SecretId 和 SecretKey。



2. 微信开发者 IDE(下载)
3. 下载名片识别小程序代码包
4. 运行环境 Node8.9 或以上

三、效果预览

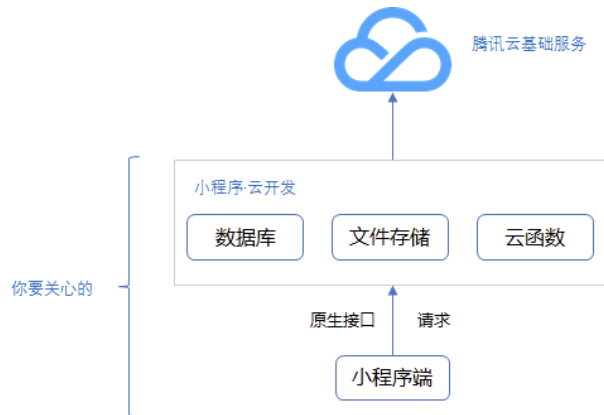


四、知识点

1. 学习如何用云开发插入、读取数据。
2. 学习如何用腾讯云的智能图像处理服务提供的 image-

ndoe-sdk 做名片识别处理。

3. 学习如何用在云开发上实现名片识别逻辑。



五、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，用于后面存储信息和开发云函数。

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，项目目录选择名片小程序Demo的目录，AppID填写已经申请公测资格的小程序对应的AppID。



2. 开通云开发环境

点击开发者工具上的【云开发】按钮

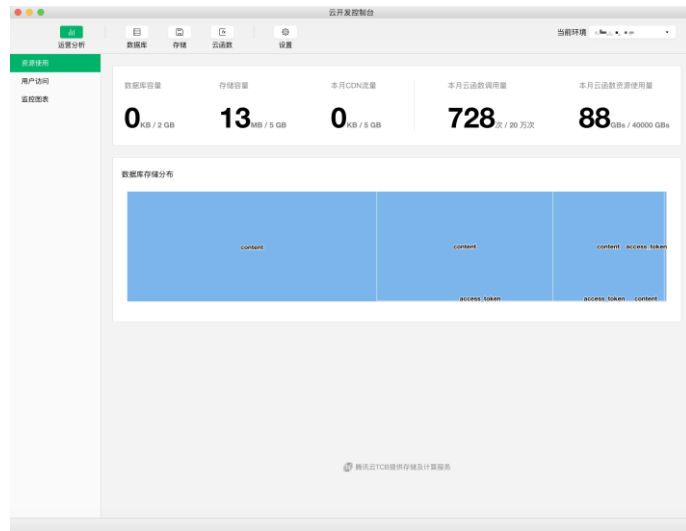


点击【同意】



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

名片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。

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



输入集合名称 "namecard"，然后点击确定即可创建集合。

The screenshot shows the '创建集合' (Create Collection) dialog box. It has a title bar '创建集合'. Below the title bar, there is a label '集合名称' (Collection Name) followed by an input field containing the text 'user'. Below the input field, there is a note '支持英文字母大小写、数字、- 和 _' (Supports English letters case, numbers, hyphen, and underscore). At the bottom right of the dialog, there are two buttons: '取消' (Cancel) and '确定' (Confirm).

六、任务二：上传名片

任务目标：利用云开发的小程序端接口上传名片图片

1. 将下面代码，输入到 client/pages/index/index.js 中的 uploadFile 方法中，用于上传图片

```
// 从相册和相机中获取图片
wx.chooseImage({ success: dRes => {
  // 展示加载组件
  wx.showLoading({ title: '上传文件',
});

let cloudPath = `${Date.now()}-${Math.floor(Math.random(0, 1) * 1000)}.png`;

// 云开发新接口：用于上传文件
wx.cloud.uploadFile({ cloudPath: cloudPath,
filePath: dRes.tempFilePaths[0], success: res => {
  if (res.statusCode < 300)
    { this.setData({fileID: res.fileID}, () => {
      // 获取临时链接
      this.getTempFileURL();});
    }
  },
  fail: err => {
    // 隐藏加载组件并提示
    wx.hideLoading();
    wx.showToast({
      title: '上传失败',
      icon: 'none'
    });
  }
}),
  fail: console.error,
})
|
```

2. 将下面代码，输入到 client/pages/index/index.js 中的 getTempFileURL 方法中，用于获取图片临时链接，并且从后续的名片识别。

```
// 云开发新接口：用于获取文件的临时链接
wx.cloud.getTempFileURL({ fileList: [{
  fileID: this.data.fileID,
}],
}).then(res => {
  console.log('获取成功', res); let files = res.fileList;

  if (files.length) { this.setData({
    coverImage: files[0].tempFileURL
  }, () => {
    this.parseNameCard();
  });
  }
  else {
    wx.showToast({
      title: '获取图片链接失败', icon: 'none'
    });
  }
}).catch(err => {
  console.error('获取失败', err); wx.showToast({
    title: '获取图片链接失败', icon: 'none'
  });
  wx.hideLoading();
});
|
```


3. 将下面代码，输入到 client/pages/index/index.js 中的 addNameCard 方法中，用于将识别并处理好的名片数据插入数据库。

```
const data = this.data
const formData = e.detail.value;
wx.showLoading({ title: '添加中'
});
formData.cover = this.data.fileID;
const db = wx.cloud.database(); db.collection('namecard').add({
data: formData
}).then((res) => { wx.hideLoading();
app.globalData.namecard.id = res._id;
wx.navigateTo({
url: '../detail/index'
});
// 重置数据
this.setData({ coverImage: null, fileID: null, formData: []
});
}).catch((e) => { wx.hideLoading(); wx.showToast({
title: '添加失败，请重试',
icon: 'none'
});
});
});
```

七、任务三：使用云函数识别名片

任务目标：用利用了 image-node-sdk 在云函数中识别名片

1. 在 cloud/functions/parseNameCard 目录下，运行以下命令，安装依赖：npm i -production
2. 填写腾讯云相关配置，新建 cloud/functions/parseNameCard/config/index.js，并填入腾讯云的 AppId, SecretId, SecretKey：

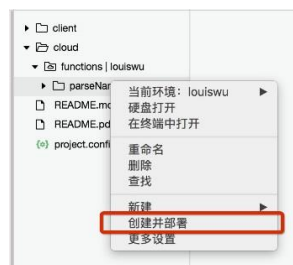
```
module.exports =
{ AppId: '',
SecretId: '', SecretKey: ''
};
```

3. 在 cloud/functions/parseNameCard/index.js 文件底部，输入以下代码：

```
const imgClient = new ImageClient
({ AppId,
  SecretId, SecretKey,
});

exports.main = async (event) => { const idCardImageUrl = event.url;
const result = await imgClient.ocrBizCard({ data: {
  url_list: [idCardImageUrl],
},
});
return JSON.parse(result.body).result_list[0];
};
```

4. 上传云函数 在微信开发者工具中，右键点击云函数 parseNameCard，选取好环境后，点击【创建并部署】。



5. 在小程序端调用云函数 parseNameCard 将下面代码，输入到 client/pages/index/index.js 中的 parseNameCard 方法中，通过此方法，调用刚刚上传的云函数处理名片。

```
wx.showLoading({ title: '解析名片', });
// 云开发新接口，调用云函数
wx.cloud.callFunction({
  name: 'parseNameCard',
  data: {
    url: this.data.coverImage
  }
}).then(res => {
  if (res.code || !res.result || !res.result.data) {
    wx.showToast({
      title: '解析失败，请重试',
      icon: 'none'
    });
    wx.hideLoading(); return;
  }

  let data = this.transformMapping(res.result.data);
  this.setData({
    formData: data
  });

  wx.hideLoading();}).catch(err => {
  console.error('解析失败，请重试。', err);
  wx.showToast({
    title: '解析失败，请重试',
    icon: 'none'
  });
  wx.hideLoading();
});
```

八、任务四：读取名片列表与详情

任务目标：利用了小程序端的接口，直接读取名片的详情或列表数据

1. 读取名片列表数据 将下面代码，输入到
client/pages/list/index.js 中的 getData 方法中，通过此方法，默认读取最多20个名片数据。

```
// 云函数新接口，用于获取数据库中数据
const db = wx.cloud.database({}); db.collection('namecard').get().then((res) => {
  let data = res.data; this.setData({
    list: data
  });
}).catch(e => {
  wx.showToast({
    title: 'db读取失败',
    icon: 'none'
  })
});
```

2. 跳转至名片详情页 将下面代码，输入到
client/pages/list/index.js 中的 getDetail 方法中，通过此方法，带上名片的 id，跳转到名片详情。

```
let _id = e.currentTarget.dataset.namecardid; app.globalData.namecard.id = _id;

wx.navigateTo({
  url: '../detail/index'
});
```

3. 读取名片详情 将下面代码，输入到
client/pages/detail/index.js 中的 getNameCardDetail 方法中，通过此方法，可以获取某个名片的详情数据。

```
// 云函数新接口，用于获取数据库中数据
const db = wx.cloud.database({});
let ncId = app.globalData.namecard.id;
db.collection('namecard').doc(ncId).get().then(res => {
  console.log('db读取成功', res.data);
  let data = res.data;
  let namecard = []; Object.keys(data).forEach((item) => {
    if (item === 'cover' || item === '_id' || item === '_openid')
      { return; }
    namecard.push({
      name: mapping[item], value: data[item]
    });
  });

  this.setData({
    cover: data.cover, namecard: namecard
  });
})
.catch(e => {
  wx.showToast({
    title: 'db读取失败', icon: 'none'
  });
});
```

九、预览

实验完成，可以在微信开发者工具中，用手机微信扫一扫预览效果。



扫码关注腾讯云「云开发」

我们一起探讨大前端 serverless 的未来