# 文件目录管理小程序的实现

1. 任务名称

文件目录管理小程序的实现

1. 任务目的

通过实现小程序端文件目录管理，实现创建目录，删除目录、获取文件信息、读取和写入文件内容等功能，让学生掌握微信小程序中文件相关api的使用，如何运用相关api解决实际问题。

1. 任务基本信息

### 1、学时

2学时。

### 2、目标

（1）熟练使用文件相关api；

（2）分析给定问题，运用微信小程序组件与api解决问题。

### 3、重难点分析

### 4、实验环境及工具

（1）开发工具：HbuilderX、微信开发者工具；

1. 任务要求

### 任务1：创建小程序并实现页面布局

要求：

1. 实现如下页面布局



图1-1邀请函导航栏

### 任务2：目录相关功能

要求:

1. 添加创建目录功能
2. 添加删除目录功能
3. 添加获取文件目录功能



图1-3目录相关功能

### 任务3：文件读写相关功能实现

要求：

1. 实现选择文件添加功能。
2. 实现获取文件信息功能
3. 读取文件内容
4. 指定文件追加内容



图1-4读写相关功能

1. 实现步骤

### 第一步：新建微信小程序

Index.wxml

<!-- 文件api -->

<view>

<view class="title">文件目录管理</view>

<view>

<view>

需要操作的目录：

<input bindinput="changeInputValue" value="{{inputValue}}" type="text" />

</view>

<view>

需要操作的文件：

<input bindinput="changeInputfileValue" value="{{inputfileValue}}" type="text" />

</view>

<view>

文件中要添加的内容：

<input bindinput="changeInputContent" value="{{inputContent}}" type="text" />

</view>

</view>

<view>

<button bindtap="addDir">创建目录</button>

<button bindtap="rmDir">删除目录</button>

<button bindtap="addFile">添加文档</button>

<button bindtap="getFileInfo">获取文件信息</button>

<button bindtap="getList">获取本地文件列表</button>

<button bindtap="readFile">读取文件内容</button>

<button bindtap="writeFile">指定文件添加文本内容</button>

</view>

<view wx:for="{{list}}" wx:key="\*this">{{item}}</view>

</view>

### 第二步：目录相关功能实现。

Index.js

var fs;//文件对象

var dirPath;//要操作的目录路径

var filePath;//要操作的文件路径

Page({

data: {

list: [],//页面渲染的文件目录列表

inputValue: '',//用户输入要操作的目录名

inputfileValue: '',//用户输入要操作的文件名

inputContent: ''//用户输入要操作的文本

},

onLoad() {

fs = wx.getFileSystemManager();

},

//用户输入目录名

changeInputValue(e) {

dirPath=wx.env.USER\_DATA\_PATH + "/" + e.detail.value;

this.setData({

inputValue: e.detail.value

})

},

//用户输入文件名

changeInputfileValue(e) {

filePath = wx.env.USER\_DATA\_PATH + "/" + e.detail.value;

this.setData({

inputfileValue: e.detail.value

})

},

//用户输入文本

changeInputContent(e) {

this.setData({

inputContent: e.detail.value

})

},

//清空输入框

clearInput() {

filePath='';

dirPath='';

this.setData({

inputValue: '',

inputfileValue: '',

inputContent:''

})

},

//获取当前文件列表

getList() {

fs.readdir({

dirPath: wx.env.USER\_DATA\_PATH,

success: res => {

this.setData({

list: res.files

})

}

})

},

//添加目录

addDir() {

fs.mkdir({

dirPath: dirPath,

success: data => {

this.getList()

this.clearInput()

},

fail: res => {

this.show("创建失败")

}

})

},

//删除目录

rmDir() {

this.check(dirPath,()=>{

fs.rmdir({

dirPath: dirPath,

success: data => {

this.getList()

this.clearInput()

},

fail: res => {

this.show("删除失败")

}

})

})

},

//检测路径是否正确

check(path, callback) {

fs.access({

path: path,

success: (res) => {

if (typeof callback == 'function') {

callback()

}

},

fail: (res) => {

//路径不存在提示用户

console.log(path)

this.show("路径不存在")

}

})

},

// 提示用户

show(title){

wx.showToast({

title: title,

icon: "none"

})

}

})

### 第三步：文件相关功能实现。

Index.js

//得到文件信息

getFileInfo() {

this.check(filePath, ()=>{

wx.getFileInfo({

filePath: filePath,

success: (res)=> {

this.show("文件大小:" + res.size)

},

fail: (res)=> {

this.show("获取文件信息失败")

}

})

})

},

addFile() { //添加文件

// 选择文件获取临时地址

wx.chooseMessageFile({

success: (res) => {

const name = res.tempFiles[0].name;

// 将文件保存至本地

fs.saveFile({

tempFilePath: res.tempFiles[0].path,

filePath: "http://store/",

success: (res) => {

// 重命名

fs.rename({

oldPath: res.savedFilePath,

newPath: wx.env.USER\_DATA\_PATH + "/" + name,

success: (res) => {

this.getList()

this.show("文件添加成功")

},

fail: (res) => {

this.show("文件添加失败")

}

})

},

fail: (res) => {

this.show("文件添加失败")

}

})

}

})

},

//读取文件

readFile() {

this.check(filePath,()=>{

fs.readFile({

filePath: filePath,

encoding: "utf-8",

success:(res)=> {

this.setData({

list:[res.data]

})

this.show("内容读取成功")

},

fail:(res)=> {

this.show("内容读取失败")

}

})

})

},

//向文件追加内容

writeFile() {

this.check(filePath,()=>{

fs.appendFile({

filePath: filePath,

data: this.data.inputContent,

encoding: "utf-8",

success:(res) =>{

this.readFile()

this.show("内容添加成功")

},

fail:(res)=> {

this.show("内容添加失败")

}

})

})

},

### 第四步：调试、运行程序。检查显示结果是否正确

1. 任务提交

按要求完成作业，根据作业报告模板给出项目中的所有页面运行结果截图，并以压缩包形式提交作业代码命名为：“学号-姓名-作业名称.zip”。