

# uniapp 安卓FTP上传下载操作原生插件

## 目 录

|                       |     |
|-----------------------|-----|
| 使用说明 .....            | 1   |
| 使用方法 .....            | 1.1 |
| 更新日志 .....            | 1.2 |
| 状态结果定义 .....          | 1.3 |
| 插件方法 .....            | 2   |
| 连接FTP .....           | 2.1 |
| 关闭连接 .....            | 2.2 |
| 获取文件列表 .....          | 2.3 |
| 创建文件夹 .....           | 2.4 |
| 上传文件 .....            | 2.5 |
| 下载文件 .....            | 2.6 |
| 删除文件/文件夹 v1.2.0 ..... | 2.7 |
| 检查是否有“所有文件”访问权限 ..... | 2.8 |
| 跳转到所有文件访问权限页面 .....   | 2.9 |

# 使用方法

---

## 介绍

安卓FTP上传下载操作原生插件，支持整个目录下载，整个目录上传，获取文件列表，上传下载进度监听等

## 联系作者

关注微信公众号可联系作者



## 插件地址

<https://ext.dcloud.net.cn/plugin?id=17041>

## 权限

1. android.permission.INTERNET
2. android.permission.WRITE\_EXTERNAL\_STORAGE
3. android.permission.READ\_EXTERNAL\_STORAGE

## 示例文件下载

如果下载的文件不是zip文件，请将扩展名改为zip即可

## 用法

在需要使用插件的页面加载以下代码

```
const module = uni.requireNativePlugin("leven-ftp-FtpModule");
```

## 页面内容

```
<template>
  <view>
    <uni-card title="FTP信息">
      <uni-section title="连接地址">
        <uni-easyinput v-model="form.address" :maxlength="-1" placeholder="请输入连
      </uni-section>
      <uni-section title="端口">
        <uni-easyinput v-model="form.port" :maxlength="-1" placeholder="请输入端口"
      </uni-section>
      <uni-section title="用户名">
        <uni-easyinput v-model="form.username" :maxlength="-1" placeholder="请输入
      </uni-section>
      <uni-section title="密码">
        <uni-easyinput v-model="form.password" :maxlength="-1" placeholder="请输入
      </uni-section>
    </uni-card>
    <uni-card title="安卓FTP远程操作原生插件">
      <button type="primary" @click="connect">连接FTP</button>
      <button type="primary" @click="close">关闭连接</button>
      <button type="primary" @click="getFiles">获取文件列表</button>
      <button type="primary" @click="mkdirs">创建文件夹</button>
      <button type="primary" @click="upload">上传文件</button>
      <button type="primary" @click="download">下载文件</button>
      <button type="primary" @click="logStr = ''">清空日志</button>
    </uni-card>
    <view>
      <uni-card class="uni-card-box" title="日志">
        <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></view>
      </uni-card>
    </view>
  </view>
</template>

<script>
  const module = uni.requireNativePlugin("leven-ftp-FtpModule");
  export default {
    data() {
      return {
        logStr: "",
        form: {
```

```

        //连接地址
        address: "",
        //端口
        port: 21,
        //用户名
        username: "",
        //密码
        password: ""
    }
}
},
methods: {
    // 连接
    connect() {
        module.connect(this.form, res => {
            this.writeLog(JSON.stringify(res))
        });
    },
    // 关闭连接
    close() {
        module.close(res => {
            this.writeLog(JSON.stringify(res))
        });
    },
    // 获取文件列表
    getFiles() {
        module.getFiles({
            path: "/测试目录"
        }, res => {
            this.writeLog(JSON.stringify(res))
        });
    },
    // 创建文件夹
    mkdirs() {
        module.mkdirs({
            path: "/测试目录/doc"
        }, res => {
            this.writeLog(JSON.stringify(res))
        });
    },
    // 上传文件
    upload() {
        module.upload({
            //本地源文件（目录/文件）
            src: "/storage/emulated/0/Pictures",
            //服务器目标目录
            target: "/测试目录"
        }, res => {
            this.writeLog(JSON.stringify(res))
        });
    }
}

```

```
    },  
    // 下载文件  
    download() {  
        module.download({  
            //服务器的文件 (目录/文件)  
            src: "/测试目录/Huawei",  
            //本地目标文件夹  
            target: "/storage/emulated/0/leven_file/Huawei"  
        }, res => {  
            this.writeLog(JSON.stringify(res))  
        });  
    },  
    // 写日志  
    writeLog(str) {  
        console.log(str)  
        let logStr = uni.$lv.date.format(null, "yyyy-mm-dd hh:MM:ss") + " " + str  
        // let logStr = str + "\n";  
        this.logStr = logStr + this.logStr;  
    }  
}  
}  
</script>  
  
<style>  
  
</style>
```

# 更新日志

---

2025-05-06 v1.2.0

- [新增] 新增【[删除文件/文件夹](#)】

2024-09-26 v1.1.3

- [优化] 优化下载文件

2024-06-21 v1.1.1

- [优化] 优化上传文件和下载文件
- [新增] 上传文件新增参数 `filename` 上传后的文件名

2024-05-07 v1.1.0

- [优化] 优化文件上传
- [新增] 新增接口【[检查是否有“所有文件”访问权限](#)】
- [新增] 新增接口【[跳转到所有文件访问权限页面](#)】

2024-03-15

首次发布

# 状态结果定义

| 状态值  | 描述        |
|------|-----------|
| 0    | 成功        |
| -100 | 权限申请未全部统一 |
| -101 | FTP已连接    |
| -102 | FTP未连接    |
| -103 | 切换文件夹失败   |
| -104 | 文件夹已存在    |
| -105 | 创建文件夹失败   |
| -106 | 本地文件不存在   |
| -107 | 服务器文件不存在  |

# 连接FTP

## 方法名

connect

## 用法

- 用法如下：

```
module.connect({
  //连接地址
  address: "",
  //端口
  port: 21,
  //用户名
  username: "",
  //密码
  password: ""
}, res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

| 参数名      | 参数类型    | 是否必填 | 默认值 | 参数描述  |
|----------|---------|------|-----|-------|
| address  | String  | 是    | 无   | 服务器地址 |
| port     | Integer | 否    | 21  | 服务器端口 |
| username | String  | 是    | 无   | 用户名   |
| password | String  | 是    | 无   | 密码    |

## 回调

- 示例

```
{
  "data": {},
  "message": "连接成功",
}
```



连接FTP

```
"code": 0
}
```

- 回调说明：

| 参数名     | 参数类型    | 参数描述            |
|---------|---------|-----------------|
| message | String  | 消息提示            |
| data    | Object  | 数据对象            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 关闭连接

## 方法名

close

## 用法

- 用法如下：

```
module.close(res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明  
无

## 回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

| 参数名     | 参数类型    | 参数描述            |
|---------|---------|-----------------|
| message | String  | 消息提示            |
| data    | Object  | 数据对象            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 获取文件列表

## 方法名

getFiles

## 用法

- 用法如下：

```
module.getFiles({
  path: "/测试目录"
}, res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

| 参数名  | 参数类型   | 是否必填 | 默认值 | 参数描述    |
|------|--------|------|-----|---------|
| path | String | 是    | 无   | 服务器目录路径 |

## 回调

- 示例

```
{
  "data": {
    "list": [
      {
        "isDirectory": true,
        "user": "1001",
        "timestamp": 1710490500000,
        "group": "www",
        "type": 1,
        "name": "Huawei",
        "isUnknown": false,
        "size": 4096,
        "isFile": false
      },
      {
        "isDirectory": true,
```

```
    "user": "1001",
    "timestamp": 1710471360000,
    "group": "www",
    "type": 1,
    "name": "a",
    "isUnknown": false,
    "size": 4096,
    "isFile": false
  },
  {
    "isDirectory": false,
    "user": "1001",
    "timestamp": 1710404760000,
    "group": "www",
    "type": 0,
    "name": "system_log.txt",
    "isUnknown": false,
    "size": 4068,
    "isFile": true
  }
]
},
"message": "",
"code": 0
}
```

- 回调说明：

| 参数名                   | 参数类型    | 参数描述       |
|-----------------------|---------|------------|
| message               | String  | 消息提示       |
| data                  | Object  | 数据对象       |
| data.list             | Array   | 数据集合       |
| data.list.isDirectory | Boolean | 是否是目录      |
| data.list.user        | String  | 所属用户       |
| data.list.timestamp   | Integer | 创建时间，单位：毫秒 |
| data.list.group       | String  | 所属组        |
| data.list.type        | Integer | 类型         |
| data.list.name        | String  | 文件名称       |
| data.list.isUnknown   | Boolean | 是否是未知      |
| data.list.size        | Integer | 文件大小，单位：字节 |

获取文件列表

| 参数名              | 参数类型    | 参数描述            |
|------------------|---------|-----------------|
| data.list.isFile | Boolean | 是否是文件           |
| code             | Integer | 返回类型，0.成功，其他：失败 |

# 创建文件夹

## 方法名

mkdirs

## 用法

- 用法如下：

```
module.mkdirs({
  path: "/测试目录/doc"
}, res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

| 参数名  | 参数类型   | 是否必填 | 默认值 | 参数描述    |
|------|--------|------|-----|---------|
| path | String | 是    | 无   | 服务器目录路径 |

## 回调

- 示例

```
{
  "data": {},
  "message": "",
  "code": 0
}
```

- 回调说明：

| 参数名     | 参数类型    | 参数描述            |
|---------|---------|-----------------|
| message | String  | 消息提示            |
| data    | Object  | 数据对象            |
| code    | Integer | 返回类型，0.成功，其他：失败 |

# 上传文件

## 方法名

upload

## 用法

- 用法如下：

```
module.upload({
  //本地源文件（目录/文件）
  src: "/storage/emulated/0/Pictures/levenFtp/测试目录/Huawei",
  // src: "/storage/emulated/0/Documents/1.docx",
  //服务器目标目录
  target: "/测试目录/2024-6-18",
  //上传后新的文件名，需带上后缀，仅上传单个文件有效，文件夹无效
  filename: "2.docx"
}, res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明

| 参数名                | 参数类型   | 是否必填 | 默认值 | 参数描述                           |
|--------------------|--------|------|-----|--------------------------------|
| src                | String | 是    | 无   | 本地目录或文件绝对路径                    |
| target             | String | 是    | 无   | 服务器目录                          |
| filename<br>v1.1.1 | String | 否    | 无   | 上传后新的文件名，需带上后缀，仅上传单个文件有效，文件夹无效 |

## 回调

- 示例

```
{
  "data": {
    "status": 1,
    "target": "/测试目录/Pictures/HwThemes/Backup",
    "type": "uploading",
    "progress": 0.9944560784274593,
    "path": "/storage/emulated/0/Pictures/HwThemes/Backup/backup_1625878742372",
  },
  "message": "",
  "code": 0
}
```

回调说明：

| 参数名           | 参数类型    | 参数描述                           |
|---------------|---------|--------------------------------|
| message       | String  | 消息提示                           |
| data          | Object  | 数据对象                           |
| data.status   | Integer | 当前上传的状态，1.上传中，2.上传成功           |
| data.target   | String  | 当前当前正在上传到的服务器目录                |
| data.type     | String  | 状态，uploading：上传中，complete：上传结束 |
| data.progress | String  | 当前文件上传的进度                      |
| data.path     | String  | 正在上传的本地文件路径                    |
| code          | Integer | 返回类型，0.成功，其他：失败                |



# 下载文件

## 方法名

download

## 用法

- 用法如下：

```

module.download({
  //服务器的文件（目录/文件）
  src: "/测试目录/Huawei",
  //本地目标文件夹
  target: "/storage/emulated/0/leven_file/Huawei"
}, res => {
  this.writeLog(JSON.stringify(res))
});
    
```

- 参数说明

| 参数名    | 参数类型   | 是否必填 | 默认值 | 参数描述          |
|--------|--------|------|-----|---------------|
| src    | String | 是    | 无   | 服务器目录或文件的绝对路径 |
| target | String | 是    | 无   | 本地目录绝对路径      |

## 回调

- 示例

```

{
  "data": {
    "status": 1,
    "target": "/storage/emulated/0/leven_file/Huawei/我的文件/龟兔赛跑.pptx",
    "type": "downloading",
    "progress": 0.9931724185827081,
    "path": "/测试目录/Huawei/我的文件/龟兔赛跑.pptx"
  },
  "message": ""
}
    
```

下载文件

```
"code": 0
}
```

- 回调说明：

| 参数名           | 参数类型    | 参数描述                             |
|---------------|---------|----------------------------------|
| message       | String  | 消息提示                             |
| data          | Object  | 数据对象                             |
| data.status   | Integer | 当前下载的状态，1.下载中，2. 下载成功            |
| data.target   | String  | 当前正在下载到本地的文件路径                   |
| data.type     | String  | 状态，downloading：下载中，complete：下载结束 |
| data.progress | String  | 当前文件下载的进度                        |
| data.path     | String  | 正在下载的服务器文件路径                     |
| code          | Integer | 返回类型，0.成功，其他：失败                  |

# 删除文件/文件夹 v1.2.0

## 方法名

deleteFile

v1.2.0

## 用法

- 用法如下：

```
module.deleteFile({
  src: "/测试目录/2024-6-18/Huawei"
}, res => {
  this.writeLog(JSON.stringify(res))
})
```

- 参数说明

| 参数名 | 参数类型   | 是否必填 | 默认值 | 参数描述          |
|-----|--------|------|-----|---------------|
| src | String | 是    | 无   | 服务器目录或文件的绝对路径 |

## 回调

- 示例

```
{
  "data": {
    "status": "onProgress",
    "path": "/测试目录/2024-6-18/Huawei/我的文件/龟兔赛跑.pptx"
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

| 参数名     | 参数类型   | 参数描述 |
|---------|--------|------|
| message | String | 消息提示 |

| 参数名         | 参数类型    | 参数描述                                               |
|-------------|---------|----------------------------------------------------|
| data        | Object  | 数据对象                                               |
| data.status | Integer | 当前进行状态，onProgress：<br>正在删除文件，onComplete：<br>所有操作完成 |
| data.path   | String  | 正在删除的服务器文件路径                                       |
| code        | Integer | 返回类型，0.成功，其他：失败                                    |

# 检查是否有“所有文件”访问权限

## 方法名

checkAllFilesPermission

## 用法

- 用法如下：

```
module.checkAllFilesPermission(res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明  
无

## 回调

- 示例

```
{
  "data": {
    "status": true
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

| 参数名         | 参数类型    | 参数描述                    |
|-------------|---------|-------------------------|
| message     | String  | 消息提示                    |
| data        | Object  | 数据对象                    |
| data.status | Boolean | 开启状态，true：已开启，false：未开启 |
| code        | Integer | 返回类型，0.成功，其他：失败         |

# 跳转到所有文件访问权限页面

## 方法名

toAllFilesPermissionPage

## 用法

- 用法如下：

```
module.toAllFilesPermissionPage(res => {
  this.writeLog(JSON.stringify(res))
});
```

- 参数说明  
无

## 回调

- 示例

```
{
  "data": {
    "status": true
  },
  "message": "",
  "code": 0
}
```

- 回调说明：

| 参数名         | 参数类型    | 参数描述                    |
|-------------|---------|-------------------------|
| message     | String  | 消息提示                    |
| data        | Object  | 数据对象                    |
| data.status | Boolean | 开启状态，true：已开启，false：未开启 |
| code        | Integer | 返回类型，0.成功，其他：失败         |