

uniapp 安卓来电去电监听原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
常见问题	1.3
插件方法	2
检测是否开启通话自动录音	2.1
跳转到开启通话自动录音页面	2.2
获取所有的录音文件	2.3
根据条件获取录音文件 v1.7.0	2.4
注册通话监听	2.5
取消通话监听	2.6
挂断电话 v1.1.0	2.7
获取所有通话列表记录 v1.2.0	2.8
根据条件获取通话列表记录 v1.4.0	2.9
拨打电话 v1.3.0	2.10
注册短信监听 v1.4.0	2.11
取消短信监听 v1.4.0	2.12
是否是双卡 v1.5.0	2.13
获取手机号 v1.6.0	2.14
移动文件 v1.6.0	2.15
复制文件 v1.6.0	2.16
删除文件或目录 v1.7.8	2.17
获取当前通话状态 v1.7.0	2.18
检查“所有文件”访问权限 v1.7.0	2.19
跳转到所有文件访问权限页面 v1.7.0	2.20
获取音频时长 v1.7.7	2.21
接听电话 v1.8.0	2.22
判断是否已注册电话监听 v1.9.0	2.23
静默发送短信 v1.9.0	2.24
获取短信列表 v1.9.0	2.25
获取手机通讯录 v1.9.0	2.26
添加手机通讯录 v1.9.1	2.27
是否有前台通知权限 v1.10.0	2.28
跳转到通知权限页面 v1.10.0	2.29

开启前台通知服务 v1.10.0	2.30
关闭前台通知服务 v1.10.0	2.31

使用方法

介绍

安卓电话通话监听原生插件，支持获取所有录音文件，来电监听，去电监听，挂断电话后返回通话录音文件

注意事项

- 当前插件2.*版本已发布，同时支持uniapp和uniappx项目
- 如果您的项目是uniapp项目建议您使用当前版本价格相对较低
- 如果您的项目是uniappx项目请使用2.*版本
- 如果您在使用过程中遇到问题可以先查看【常见问题】，如果还没有解决可以扫描下方二维码联系作者，作者将全力协助您使用插件
- 两个版本的插件作者都会维护可放心使用

联系作者

关注微信公众号可联系作者



插件适配的机型

1. 华为
2. 荣耀
3. vivo
4. oppo

使用方法

5. 小米
6. 魅族
7. 一加
8. 联想
9. 中兴
10. 摩托罗拉
11. 三星

需要适配其他机型，请联系作者

插件地址

<https://ext.dcloud.net.cn/plugin?id=16275>

权限

1. android.permission.ANSWER_PHONE_CALLS
2. android.permission.READ_PHONE_STATE
3. android.permission.READ_PHONE_NUMBERS
4. android.permission.CALL_PHONE
5. android.permission.READ_CALL_LOG
6. android.permission.WRITE_CALL_LOG
7. android.permission.RECORD_AUDIO
8. android.permission.WRITE_EXTERNAL_STORAGE
9. android.permission.READ_EXTERNAL_STORAGE
10. android.permission.RECEIVE_SMS
11. android.permission.SEND_SMS
12. android.permission.READ_SMS
13. android.permission.SYSTEM_ALERT_WINDOW
14. android.permission.SYSTEM_OVERLAY_WINDOW
15. android.permission.READ_CONTACTS
16. android.permission.WRITE_CONTACTS

示例文件和演示程序下载

使用方法

加QQ群下载，群号：106179987



uniapp leven 系列...

群号：106179987



扫一扫二维码，入群聊

用法

在需要使用插件的页面加载以下代码

```
const module = uni.requireNativePlugin("leven-call-CallModule");
```

页面内容

```
<template>
  <view>
    <uni-card title="安卓电话通话监听原生插件">
      <button type="primary" @click="checkCallAutoRecorder">检测是否开启通话自动录音</button>
      <button type="primary" @click="toCallAutoRecorderPage">跳转到开启通话自动录音页面</button>
      <button type="primary" @click="getAllRecorderFiles">获取所有的录音文件</button>
      <button type="primary" @click="registerListener">注册通话监听</button>
      <button type="primary" @click="unRegisterListener">取消通话监听</button>
      <button type="primary" @click="logStr = ''">清空日志</button>
    </uni-card>
  </view>
  <view>
    <uni-card class="uni-card-box" title="日志">
```

```

    <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></\
    </uni-card>
  </view>
</view>
</template>

<script>
  const module = uni.requireNativePlugin("leven-call-CallModule");
  export default {
    data() {
      return {
        logStr: "",
      }
    },
    methods: {
      // 检测是否开启通话自动录音
      checkCallAutoRecorder() {
        module.checkCallAutoRecorder(res => {
          this.writeLog(JSON.stringify(res))
        });
      },
      // 跳转到开启通话自动录音页面
      toCallAutoRecorderPage() {
        module.toCallAutoRecorderPage(res => {
          this.writeLog(JSON.stringify(res))
        });
      },
      // 获取所有录音文件
      getAllRecorderFiles() {
        module.getAllRecorderFiles(res => {
          this.writeLog(JSON.stringify(res))
        });
      },
      //注册通话监听
      registerListener() {
        module.registerListener(res => {
          this.writeLog(JSON.stringify(res))
        });
      },
      //取消通话监听
      unregisterListener() {
        module.unregisterListener(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";
      }
    }
  }

```

使用方法

```
        this.logStr = logStr + this.logStr;
    }
}
</script>

<style>

</style>
```

更新日志

- [优化] 适配华为liantong系列手机
- [优化] 优化返回通话状态的唯一标识
- [优化] 新增返回当前通话流程状态的唯一标识
- [优化] 通话监听新增返回参数 录音文件的创建时间等
- [优化] 静默发送短信新增发送状态回调
- [优化] 优化通话监听回调不返回手机号问题
- [优化] 新增回调通话状态6(通话中时来电占线)
- [优化] 优化通话监听回调有时不返回通话文件路径问题
- [优化] 获取电话号码返回结果新增所在卡槽id
- [新增] 新增方法【[是否有前台通知权限](#)】
- [新增] 新增方法【[跳转到通知权限页面](#)】
- [新增] 新增方法【[开启前台通知服务](#)】
- [新增] 新增方法【[关闭前台通知服务](#)】
- [优化] 优化vivo检测是否开启通话自动录音
- [优化] 优化中兴部分手机通话录音获取不到的问题
- [优化] 优化适配三星系列机型
- [新增] 新增方法【[添加手机通讯录](#)】
- [优化] 优化重复注册注册监听问题
- [优化] 优化适配华为畅享30e机型
- [新增] 新增接口【[判断是否已注册电话监听](#)】
- [新增] 新增接口【[获取通讯录](#)】
- [新增] 新增接口【[获取短信列表](#)】
- [新增] 新增接口【[静默发送短信](#)】
- [优化] 优化vivo安卓10获取是否开启自动录音失败的bug
- [修复] 修复来电不显示来电号码的bug

更新日志

- [新增] 新增接口【[接听电话](#)】
- [优化] 适配魅族系列部分录音路径
- [优化] 新增方法【[删除文件或目录](#)】
- [优化] 新增方法【[获取音频时长](#)】

*[优化] 适配部分vivo系列手机获取不到录音问题

*[优化] 适配摩托罗拉系列手机

- [优化] 拨打电话新增是否是座机参数
- [优化] 适配oppo老手机录音文件路径
- [优化] 适配联想、一加手机系列
- [优化] 优化通话记录根据条件查找可以多条件查询
- [优化] 通话记录返回结果新增卡槽索引
- [优化] 获取录音文件改为在子线程中查询防止文件过多导致应用无响应
- [新增] 新增接口【[根据条件搜索录音文件](#)】
- [新增] 新增接口【[获取当前通话状态](#)】
- [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](#)】
- [优化] 查找通话记录支持通过时间日期查询
- [优化] 通话和响铃时打开屏幕常亮，通话结束关闭屏幕常亮功能
- [新增] 新增接口【[获取手机号](#)】
- [新增] 新增接口【[移动文件](#)】
- [新增] 新增接口【[复制文件](#)】
- [新增] 新增接口【[是否是双卡](#)】
- [优化] [注册电话监听](#)新增返回状态：5.去电对方拒绝接听或未接听
- [新增] 新增接口【[注册短信监听](#)】
- [新增] 新增接口【[取消短信监听](#)】
- [新增] 新增接口【[根据条件获取通话记录](#)】
- [优化] 通话监听新增状态：4.未接或拒绝接听（仅限来电）
- [新增] 新增接口【[拨打电话](#)】

更新日志

- [新增] 新增接口【[获取所有通话记录](#)】
- [优化] 修复通话监听重复返回数据的bug
- [优化] 获取所有录音文件新增返回参数
- [新增] 新增接口【[挂断电话](#)】
- [优化] 修复荣耀手机安卓系统获取数据失败的bug
- [优化] 优化vivo手机获取数据失败的bug
- [优化] 优化oppo手机获取是否开启通话自动录音获取不到的bug
- [优化] 优化oppo手机跳转到通话自动录音页面无响应的bug

首次发布

常见问题

有些小米手机需要关闭空白通行证才能监听通话状态

如果没有返回录音文件路径请通过以下方法检查：

- 1.检查手机是否打开了通话自动录音功能
- 2.安卓11以下的手机检查是否开启了文件读写权限，安卓11以上(包含安卓 11)检查是否打开了所有文件访问权限功能
- 3.安卓低版本可能还需要开启“应用内安装其它应用权限”才能获取到录音文件

录音文件路径为系统目录有访问权限限制，需要将录音文件复制到内部存储路径中才能上传，可调用【[复制文件](#)】或【[移动文件](#)】转移到内部存储目录中，通过内部存储的录音文件路径上传即可

需要开启悬浮窗权限

锁屏监听数据需[开启前台通知服务](#)

手机生成的录音文件为系统生成，可能不是标准的音频文件，上传到后台会播放不了，不过您不用担心，作者有音视频转换的插件，插件地址：<https://ext.dcloud.net.cn/plugin?id=16523>，文档地址：

<https://z7poo9xpe4.k.topthink.com/@l4rz1dgpx/shiyongfangfa.html>

您可以调用音视频转换的接口完成格式转换

检测是否开启通话自动录音

方法名

checkCallAutoRecorder

用法

- 用法如下：

```
module.checkCallAutoRecorder(res => {  
  console.log(res)  
});
```

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "isOpen": 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.isOpen	Boolean	true：已开启自动录音，false：未开启自动录音
code	Integer	返回类型，0.成功，其他：失败

跳转到开启通话自动录音页面

方法名

toCallAutoRecorderPage

用法

- 用法如下：

```
module.toCallAutoRecorderPage(res => {
  console.log(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

获取所有的录音文件

方法名

getAllRecorderFiles

用法

- 用法如下：

```
module.getAllRecorderFiles(res => {  
  console.log(res)  
});
```

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "list": [  
      "/storage/emulated/0/Sounds/CallRecord/10 101035_20231020145025.m4a",  
      "/storage/emulated/0/Sounds/CallRecord/132 9797 1070_20231020165355.m4a",  
      "/storage/emulated/0/Sounds/CallRecord/165 1006 6375_20231021142216.m4a"  
    ],  
    "files": [{  
      "lastModifiedDate": "2024-01-26 15:07:40"  
      "time": 1706252857000,  
      "lastModifiedTime": 1706252860000  
      "timeDate": "2024-01-26 15:07:37"  
      "path": "/storage/emulated/0/Sounds/CallRecord/178 0338 5822 2024012615"  
    }]  
  },  
  "message": "",  
  "code": 0  
}
```

- 回调说明：

获取所有的录音文件

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	录音文件路径列表
data.files v1.2.0	Array	录音文件列表
data.files.lastModifiedTimeDate	String	最后修改时间的格式化数据
data.files.time	Integer	添加时间，时间戳
data.files.lastModifiedTime	Integer	最后修改时间，时间戳
data.files.timeDate	String	添加时间的格式化数据
data.files.path	String	文件路径
code	Integer	返回类型，0.成功，其他：失败

根据条件获取录音文件 v1.7.0

方法名

getRecorderFiles

v1.7.0

用法

- 用法如下：

```
module.getRecorderFiles({
  //录音时间段
  times: [1711209600000, 1711296000000]
}, 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
times	Array	是	无	录音文件的创建 时间，格式:[开始 时间,结束时间]， 时间单位：毫秒

回调

- 示例

```
{
  "data": {
    "list": [
      "/storage/emulated/0/Sounds/CallRecord/10 101035_20231020145025.m4a",
      "/storage/emulated/0/Sounds/CallRecord/132 9797 1070_20231020165355.m4a"
      "/storage/emulated/0/Sounds/CallRecord/165 1006 6375_20231021142216.m4a"
    ],
    "files": [{
      "lastModifiedDate": "2024-01-26 15:07:40"
      "time": 1706252857000,
      "lastModifiedTime": 1706252860000
    }
  ]
}
```

```
        "timeDate": "2024-01-26 15:07:37"
        "path": "/storage/emulated/0/Sounds/CallRecord/178 0338 5822 2024012615
    }]
},
"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	录音文件路径列表
data.files	Array	录音文件列表
data.files.lastModifiedTimeDate	String	最后修改时间的格式化数据
data.files.time	Integer	添加时间，时间戳
data.files.lastModifiedTime	Integer	最后修改时间，时间戳
data.files.timeDate	String	添加时间的格式化数据
data.files.path	String	文件路径
code	Integer	返回类型，0.成功，其他：失败

注册通话监听

方法名

registerListener

通话状态会在该方法中返回，具体请参考下方的说明

录音文件不是标准格式的话可以通过音视频转换插件转换为需要的格式即可，具体可参考【[常见问题](#)】

用法

- 用法如下：

```
module.registerListener(res => {  
  console.log(res)  
});
```

- 参数说明

无

回调

- 示例

```
{  
  "data": {},  
  "message": "注册监听成功",  
  "code": 0  
}  
  
{  
  "data": {  
    "isIncoming": false,  
    "status": 2,  
    "phoneNumber": "180****2781"  
  },  
  "message": "",  
  "code": 0  
}  
  
{
```

```
"data": {
  "status": 3,
  "phoneNumber": "180****2781",
  "audioPath": "/storage/emulated/0/Sounds/CallRecord/未知@180 **** 2781_2024",
  "duration": 37739
},
"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	通话类型，1.来电响铃，2.去电或来电通话中，3.电话挂断，4.来电拒接或未接听，5.去电对方拒接或未接听，6.通话中时来电占线
data.phoneNumber	String	来电或去电的电话号码
data.audioPath	String	录音文件，status=3时如果有录音文件会返回
data.duration	Integer	录音时长，status=3时如果有录音文件会返回，单位：毫秒
data.isIncoming	Boolean	是否是来电，true：来电，false：去电
data.lastModifiedDate v1.10.8	String	录音文件最后修改时间的格式化数据，status=3时如果有录音文件会返回
data.time v1.10.8	Integer	录音文件添加时间，时间戳，status=3时如果有录音文件会返回
data.lastModifiedTime v1.10.8	Integer	录音文件最后修改时间，时间戳，status=3时如果有录音文件会返回

参数名	参数类型	参数描述
data.timeDate v1.10.8	String	录音文件添加时间的格式化数据，status=3时如果有录音文件会返回
data.uuid v1.10.12	String	当前通话流程中的唯一id
code	Integer	返回类型，0.成功，其他：失败

录音文件不是标准格式的话可以通过音视频转换插件转换为需要的格式即可，具体可参考【[常见问题](#)】

取消通话监听

方法名

unRegisterListener

用法

- 用法如下：

```
module.unRegisterListener(res => {
  console.log(res)
});
```

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "取消监听成功",
  "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

挂断电话 v1.1.0

- 用法如下：

- 参数说明

无

- 示例

- 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

获取所有通话列表记录 v1.2.0

方法名

getAllCalls v1.2.0

用法

- 用法如下：

```
module.getAllCalls(res => {  
  console.log(res)  
});
```

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "list": [{  
      "name": "我的手机号码",  
      "number": "180****2781",  
      "duration": 35,  
      "type": 1,  
      "date": "2024-01-26 12:16:37",  
      "dateTime": 1706242597486  
    }]  
  },  
  "message": "操作成功",  
  "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.name	String	通话名称
data.number	String	电话号码
data.duration	Integer	通话时长，单位：秒
data.type	Integer	通话类型，1.呼入，2.呼出，3.未接，4.语音邮件，5.拒绝接听
data.date	String	通话时间，年月日格式
data.dateTime	Integer	通话时间，时间戳格式
data.slotId	Integer	卡槽索引
code	Integer	返回类型，0.成功，其他：失败

根据条件获取通话列表记录 v1.4.0

方法名

getCalls v1.4.0

用法

- 用法如下：

```
module.getCalls({
  //条件，1.根据号码查找，2.根据名称查找，3.根据时间段查找，4.联合查询
  type: 4,
  //查找的内容，type=1和2时有效
  value: 10086,
  //根据时间段查找，type=3时有效
  times: [1711209600000, 1711296000000],
  //联合查询参数，type=4时有效
  params: {
    //查询的姓名
    name: "",
    //查询的手机号码
    number: "",
    //查询的时间段
    times: [1711814400000, 1711900799000],
  }
}, 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
type	Integer	是	无	查找类型，1.根据号码查找，2.根据名称查找，3.根据时间段查找，4.联合查询
value	Integer	是	无	查找的内容，type=1和2时有效

参数名	参数类型	是否必填	默认值	参数描述
times	Integer	否	无	时间段区间，单位：毫秒，如： [1711209600000, 1711296000000]，type=3时有效
params	Object	否	无	联合查询参数，type=4时有效
params.name	String	否	无	联合查询名称
params.number	String	否	无	联合查询手机号
params.times	String	否	无	联合查询时间段

回调

- 示例

```
{
  "data": {
    "list": [{
      "name": "我的手机号码",
      "number": "180****2781",
      "duration": 35,
      "type": 1,
      "date": "2024-01-26 12:16:37",
      "dateTime": 1706242597486,
      "slotId": 0
    }]
  },
  "message": "操作成功",
  "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.name	String	通话名称

参数名	参数类型	参数描述
data.number	String	电话号码
data.duration	Integer	通话时长，单位：秒
data.type	Integer	通话类型，1.呼入，2.呼出，3.未接，4.语音邮件，5.拒绝接听
data.date	String	通话时间，年月日格式
data.dateTime	Integer	通话时间，时间戳格式
data.slotId	Integer	卡槽索引
code	Integer	返回类型，0.成功，其他：失败

拨打电话 v1.3.0

方法名

callPhone

v1.3.0

用法

- 用法如下：

```
module.callPhone({
  // 拨打号码
  number: 10086,
  // 卡槽索引, 1.卡槽1(默认), 2.卡槽2
  slot: 1,
  //是否是座机
  isTelephone: false
}, 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
number	String	是	无	要拨打的电话号码
slot	Integer	否	1	选择的拨打的卡槽序号, 1.卡槽1, 2.卡槽2
isTelephone v1.7.3	Boolean	否	false	拨打的是否是座机

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

注册短信监听 v1.4.0

方法名

registerSmsListener v1.4.0

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{  
  "data": {},  
  "message": "注册监听成功",  
  "code": 0  
}  
  
{  
  "data": {  
    "content": "【余额查询】尊敬的客户，您好！扣除您已产生的消费(39.45元，其中本月消费39.",  
    "number": "10086"  
  },  
  "message": "收到短信",  
  "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.content	String	短信内容
data.number	String	短信发送号码
code	Integer	返回类型，0.成功，其他：失败

取消短信监听 v1.4.0

方法名

unRegisterSmsListener

用法

- 用法如下：

```
module.unRegisterSmsListener(res => {
  console.log(res)
});
```

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "取消监听成功",
  "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

是否是双卡 v1.5.0

方法名

isMultiSim

v1.5.0

用法

- 用法如下：

```
module.isMultiSim(res => {
  console.log(res)
});
```

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "isMultiSim":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.isMultiSim	Boolean	true:是双卡，false：不是双卡
code	Integer	返回类型，0.成功，其他：失败

获取手机号 v1.6.0

方法名 v1.6.0

getPhoneNumber

权限

- android.permission.READ_PHONE_STATE
- android.permission.READ_SMS
- android.permission.READ_PHONE_NUMBERS

需要在manifest.json中勾选相应的权限

用法

- 用法如下：

```
module.getPhoneNumber(res => {  
  console.log(res)  
})
```

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "numbers": [  
      {  
        "displayName": "CMCC",  
        "mnc": "02",  
        "mcc": "460",  
        "carrierName": "中国移动",  
        "number": "15912345678",
```

```
        "subId": 2,
        "slotId": 0
    },
    {
        "displayName": "中国电信",
        "mnc": "03",
        "mcc": "460",
        "carrierName": "中国电信",
        "number": "18012345678",
        "subId": 1,
        "slotId": 1
    }
]
},
"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.numbers	Array	手机号列表
data.numbers.displayName	String	显示名称
data.numbers.mnc	String	移动国家代码
data.numbers.mcc	String	移动网络代码
data.numbers.carrierName	String	运营商名称
data.numbers.number	String	手机号
data.numbers.subId	Integer	订阅id
data.numbers.slotId	Integer	所在卡槽id
v1.10.1		
code	Integer	返回类型，0.成功，其他：失败

移动文件 v1.6.0

方法名 **v1.6.0**

`moveFile`

此方法会移动成功后会删除移动前的文件

用法

- 用法如下：

```
module.moveFile({
  fromFilename: "/storage/emulated/0/qqmusic/easter_egg_config_res/easter_egg_
  toFilename: "/storage/emulated/0/leven_file/a/egg_51.png"
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
fromFilename	String	是	无	要复制的文件路径
toFilename	String	是	无	目的文件路径

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

复制文件 v1.6.0

- 用法如下：

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
fromFilename	String	是	无	要复制的文件路径
toFilename	String	是	无	目的文件路径

- 示例

- 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

删除文件或目录 v1.7.8

方法名

deleteFile

用法

- 用法如下：

```
module.deleteFile({
  filePath: "/storage/emulated/0/leven_file/a/egg_51.png"
}, res => {
  console.log(res)
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
filePath	String	是	无	文件或目录的路径

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

参数名	参数类型	参数描述
code	Integer	返回类型，0.成功，其他：失败

获取当前通话状态 v1.7.0

- 用法如下：

- 参数说明
无
- 示例

- 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.state	Integer	通话状态：0.空闲状态，1.响铃，2.通话中
code	Integer	返回类型，0.成功，其他：失败

检查“所有文件” 访问权限 v1.7.0

方法名

checkAllFilesPermission

v1.7.0

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 v1.7.0

方法名

toAllFilesPermissionPage v1.7.0

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

获取音频时长 v1.7.7

方法名

getDuration

用法

- 用法如下：

```
module.getDuration({
  path: "/storage/emulated/0/Download/WeiXin/赵雷-成都.mp3"
}, res => {
  console.log(res)
});
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
path	Integer	是	无	本地绝对路径或网络地址

回调

- 示例

```
{
  "data": {
    "duration": 445518
  },
  "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

参数名	参数类型	参数描述
data.duration	Integer	音频总时长，单位：毫秒
code	Integer	返回类型，0.成功，其他：失败

接听电话 v1.8.0

方法名

answerRingingCall v1.8.0

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "操作成功",
  "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

判断是否已注册电话监听 v1.9.0

方法名

用法

- 用法如下：

- 参数说明

无

回调

- 示例

- 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.isRegisterListener	Boolean	true：已注册监听，false：未注册监听
code	Integer	返回类型，0.成功，其他：失败

静默发送短信 v1.9.0

方法名

hideSendSms v1.9.0

用法

- 用法如下：

```
module.hideSendSms({
  //发送短信的号码
  phoneNumber: "10086",
  //发送内容
  message: "ye",
  //卡槽，默认：系统默认卡槽
  subId: 2
}, 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
phoneNumber	Integer	是	无	要发送的电话号码
message	String	是	无	发送的内容
subId	Integer	否	默认卡槽	卡槽

回调

- 示例

```
{
  "data": {
    "type": "send"
  },
  "message": "短信发送成功",
  "code": 0
}
```

```
}

{
  "data": {
    "type": "sendResult"
  },
  "message": "短信已发送",
  "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.type 1.10.7	String	发送状态，send：发送成功，sendResult：短信已送达
code	Integer	返回类型，0.成功，-101：短信发送失败，-102：短信发送无服务。-103：PDU为空，-104：无线电关闭，-105：短信未送达

获取短信列表 v1.9.0

方法名

getSms v1.9.0

注意事项

获取短信可能会有一些类似验证码的短信获取不到，这是跟手机的短信设置相关，有些手机限制了第三方读取验证码短信，您可以去短信设置页面取消这个限制就可以获取到了

用法

- 用法如下：

```
module.getSms({
  //短信内容，模糊搜索
  content: "预警",
  //发件人号码
  phoneNumber: "10086"
}, 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
content	String	否	无	搜索的短信内容
phoneNumber	String	否	无	搜索的电话号码

回调

- 示例

```
{
  "data": {
    "list": [
      {
        "type": 1,
```

```
    "content": "武汉市气象台2024年07月13日10时45分升级发布暴雨橙色预警信号：过去1小时  
    "time": 1720842599798,  
    "phoneNumber": "10086",  
    "person": 0,  
    "read": 1  
  },  
  {  
    "type": 1,  
    "content": "武汉市气象台2024年07月13日10时45分升级发布暴雨橙色预警信号：预计未来  
    "time": 1720839582562,  
    "phoneNumber": "10086",  
    "person": 0,  
    "read": 1  
  },  
  {  
    "type": 1,  
    "content": "武汉市气象台2024年07月13日00时51分发布暴雨橙色预警信号：预计未来3小时  
    "time": 1720804801516,  
    "phoneNumber": "10086",  
    "person": 0,  
    "read": 1  
  }  
]  
},  
"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.type	Integer	短信类型：1.接收，2.发送，3.草稿，4.发信箱，5.发送失败，6.待发送
data.content	String	短信内容
data.time	Integer	时间戳
data.phoneNumber	String	发送人
data.person	Integer	联系人里的序号
data.read	Integer	已读状态，0.未读，1.已读
code	Integer	返回类型，0.成功，其他：失败

获取手机通讯录 v1.9.0

方法名

getContacts v1.9.0

用法

- 用法如下：

```
module.getContacts({
  //手机号码，可以为空
  phoneNumber: "",
  //姓名
  name: "胡"
}, 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
phoneNumber	String	否	无	搜索的电话号码
name	String	否	无	搜索姓名，模糊搜索

回调

- 示例

```
{
  "data": {
    "list": [
      {
        "name": "胡1",
        "phoneNumber": "18671211111"
      },
      {
        "name": "胡2",
        "phoneNumber": "13471111111"
      }
    ]
  }
}
```

```
    }
  ]
},
"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.name	String	姓名
data.phoneNumber	String	电话
code	Integer	返回类型，0.成功，其他：失败

添加手机通讯录 v1.9.1

方法名

addContact v1.9.1

用法

- 用法如下：

```
module.addContact({
  //手机号码
  phoneNumber: "15926128990",
  //姓名
  name: "大L"
}, 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
phoneNumber	String	是	无	电话号码
name	String	是	无	姓名

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

参数名	参数类型	参数描述
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

是否有前台通知权限 v1.10.0

- 用法如下：

- 参数说明

无

- 示例

- 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	权限状态，true：已开启，false：未开启
code	Integer	返回类型，0.成功，其他：失败

跳转到通知权限页面 v1.10.0

方法名

toForegroundPage

v1.10.0

用法

- 用法如下：

```
module.toForegroundPage(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	权限状态，true：已开启，false：未开启
code	Integer	返回类型，0.成功，其他：失败

开启前台通知服务 v1.10.0

方法名

startForeground

v1.10.0

用法

- 用法如下：

```
module.startForeground({
  //前台通知服务标题
  title: "安卓电话通话监听原生插件",
  //前台通知服务内容
  content: "安卓电话通话监听原生插件正在运行中",
  //icon图标名称，不需要加后缀名，需放置到【nativeplugins】>【leven-call】>【android】
  icon: "icon"
}, 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
title	String	否	无	前台通知标题
content	String	否	无	前台通知描述
icon	String	否	无	前台通知图标

前台通知图标，不需要加后缀名，需放置到【nativeplugins】>【leven-call】>【android】>【res】>【mipmap】>图片名称

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

关闭前台通知服务 v1.10.0

方法名

stopForeground

v1.10.0

用法

- 用法如下：

```
module.stopForeground(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