

uniapp 安卓设备管理UTS原生API插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
插件方法	2
设备是否有权查看应用使用权限	2.1
设备访问应用使用权限是否打开	2.2
打开应用使用情况访问权限页面	2.3
获取应用的使用情况	2.4
获取设备安装的app	2.5
申请文件读写权限	2.6
检查是否有“所有文件”访问权限	2.7
跳转到所有文件访问权限页面	2.8
截图(应用内)	2.9
监听前台通知服务	2.10
取消监听前台通知服务	2.11
开启前台通知服务	2.12
关闭前台通知服务	2.13
开启截图服务	2.14
是否开启截图服务	2.15
截图(应用外)	2.16
屏幕状态监听	2.17
取消屏幕状态监听	2.18
是否注册屏幕状态监听	2.19
申请悬浮窗权限	2.20
是否开启悬浮窗	2.21

使用方法

介绍

安卓设备管理UTS原生API插件集成了设备中应用的使用情况查看，设备安装的所有应用列表，应用内截图和应用外截图，屏幕状态监听，悬浮窗权限等多个功能，插件UTS开发，支持uniapp x

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=19982>

用法

在需要使用插件的页面加载以下代码

```
import * as module from "@uni_modules/leven-uts-equipment"
```

页面内容

```
<template>
  <view>
    <uni-card title="安卓设备管理原生插件">
```

```

<button type="primary" @click="hasUsageSetting">设备是否有权查看应用使用权限</button>
<button type="primary" @click="isOpenUsageSetting">设备访问应用使用权限是否打开</button>
<button type="primary" @click="openUsageSettingPage">打开应用使用情况访问权限页面</button>
<button type="primary" @click="getAppsUse">获取应用的使用情况</button>
<button type="primary" @click="getInstallApps">获取设备安装的app</button>
<button type="primary" @click="filePermission">申请文件读写权限</button>
<button type="primary" @click="checkAllFilesPermission">检查是否有“所有文件”访问权限</button>
<button type="primary" @click="toAllFilesPermissionPage">跳转到所有文件访问权限页面</button>
<button type="primary" @click="screenShot">截图(应用内)</button>
<button type="primary" @click="setForegroundServiceListener">监听前台通知服务</button>
<button type="primary" @click="clearForegroundServiceListener">取消监听前台通知服务</button>
<button type="primary" @click="startForegroundService">开启前台通知服务</button>
<button type="primary" @click="stopForegroundService">关闭前台通知服务</button>
<button type="primary" @click="openScreenShortService">开启截图服务(应用外)</button>
<button type="primary" @click="isOpenScreenShortService">是否开启截图服务</button>
<button type="primary" @click="screenShotWindow">截图(应用外)</button>
<button type="primary" @click="screenListener">屏幕状态监听</button>
<button type="primary" @click="removeScreenListener">取消屏幕状态监听</button>
<button type="primary" @click="isScreenListener">是否注册屏幕状态监听</button>
<button type="primary" @click="requestAlertWindow">申请悬浮窗权限</button>
<button type="primary" @click="isAlertWindow">是否开启悬浮窗</button>
<button type="primary" @click="logStr = ''">清空日志</button>
</uni-card>
<uni-card class="uni-card-box" title="日志">
  <view><text style="font-size: 14px; flex-wrap: wrap;">{{logStr}}</text></view>
</uni-card>
</view>
</template>

<script>
import * as module from "@uni_modules/leven-uts-equipment"
export default {
  data() {
    return {
      logStr: "",
    }
  },
  methods: {
    //判断当前设备是否有权查看使用权限用
    hasUsageSetting() {
      module.hasUsageSetting(res => {
        this.writeLog("判断当前设备是否有权查看使用权限用：" + JSON.stringify(res))
      });
    },
    //判断当前设备访问应用使用权限是否打开
    isOpenUsageSetting() {
      module.isOpenUsageSetting(res => {
        this.writeLog("判断当前设备访问应用使用权限是否打开：" + JSON.stringify(res))
      });
    },
  },
}

```

```

//开启应用使用情况访问权限
openUsageSettingPage() {
  module.openUsageSettingPage(res => {
    this.writeLog("开启应用使用情况访问权限：" + JSON.stringify(res))
  });
},
//获取应用的使用情况
getAppsUse() {
  module.getAppsUse({
    //获取多少天内的使用情况，0.当天，默认：0
    dayNumber: 0
  }, res => {
    this.writeLog("获取应用的使用情况：" + JSON.stringify(res))
  });
},
//获取设备安装的app
getInstallApps() {
  module.getInstallApps(res => {
    this.writeLog("获取设备安装的app：" + JSON.stringify(res))
  });
},
//申请文件权限
filePermission() {
  module.filePermission(res => {
    this.writeLog("申请文件权限：" + JSON.stringify(res))
  });
},
//检查是否有“所有文件”访问权限
checkAllFilesPermission() {
  module.checkAllFilesPermission(res => {
    this.writeLog("检查是否有“所有文件”访问权限：" + JSON.stringify(res))
  });
},
//跳转到所有文件访问权限页面
toAllFilesPermissionPage() {
  module.toAllFilesPermissionPage(res => {
    this.writeLog("跳转到所有文件访问权限页面：" + JSON.stringify(res))
  });
},
//截图（应用内）
screenShot() {
  module.screenShot(res => {
    this.writeLog("截图（应用内）" + JSON.stringify(res))
  });
},
//监听前台服务通知
setForegroundServiceListener() {
  module.setForegroundServiceListener(res => {
    this.writeLog("监听前台服务通知：" + JSON.stringify(res))
  });
}

```

```

},
//取消监听前台通知服务
clearForegroundServiceListener() {
  module.clearForegroundServiceListener(res => {
    this.writeLog("取消监听前台通知服务:" + JSON.stringify(res))
  });
},
//开启前台通知服务(应用外截图必需)
startForegroundService() {
  //设置前台通知图标请参考插件文档
  module.startForegroundService({
    title: "前台通知标题",
    description: "前台通知描述"
  }, res => {
    this.writeLog("开启前台通知服务:" + JSON.stringify(res))
  })
},
//关闭前台通知服务
stopForegroundService() {
  module.stopForegroundService(res => {
    this.writeLog("关闭前台通知服务:" + JSON.stringify(res))
  });
},
//开启截图服务(应用外截图必需)
openScreenShortService() {
  module.openScreenShortService(res => {
    this.writeLog(JSON.stringify(res))
  });
},
//是否开启截图服务
isOpenScreenShortService() {
  module.isOpenScreenShortService(res => {
    this.writeLog(JSON.stringify(res))
  });
},
//截图(应用外)
screenShotWindow() {
  //这里做延迟是为了测试方便,正式环境中可以去掉延迟
  setTimeout(() => {
    module.screenShotWindow(res => {
      this.writeLog(JSON.stringify(res))
    });
  }, 5000)
},
//屏幕状态监听
screenListener() {
  module.screenListener(res => {
    this.writeLog(JSON.stringify(res))
  });
},

```

```

//取消屏幕状态监听
removeScreenListener() {
  module.removeScreenListener(res => {
    this.writeLog(JSON.stringify(res))
  });
},
//是否注册屏幕状态监听
isScreenListener() {
  module.isScreenListener(res => {
    this.writeLog(JSON.stringify(res))
  });
},
//申请悬浮窗权限
requestAlertWindow() {
  module.requestAlertWindow(res => {
    this.writeLog(JSON.stringify(res))
  });
},
//是否开启悬浮窗
isAlertWindow() {
  module.isAlertWindow(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

2024-08-26

首次发布

设备是否有权查看应用使用权限

方法名

hasUsageSetting

用法

- 用法如下：

```
module.hasUsageSetting(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	是否有查看权限
code	Integer	返回类型，0.成功，其他：失败

设备访问应用使用权限是否打开

方法名

isOpenUsageSetting

用法

- 用法如下：

```
module.isOpenUsageSetting(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	应用访问权限是否打开
code	Integer	返回类型，0.成功，其他：失败

打开应用使用情况访问权限页面

方法名

openUsageSettingPage

用法

- 用法如下：

```
module.openUsageSettingPage(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	应用访问权限是否打开
code	Integer	返回类型，0.成功，其他：失败

获取应用的使用情况

方法名

getAppsUse

用法

- 用法如下：

```
module.getAppUse({
  //获取多少天内的使用情况，0.当天，默认：0
  dayNumber: 0
}, res => {
  this.writeLog("获取应用的使用情况：" + 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
dayNumber	Integer	否	0	获取多少天前的使用情况

回调

- 示例

```
{
  "data": {
    "list": [
      {
        "usedCount": 1,
        "packageName": "uni.UNIB291F5B",
        "icon": "iVBORw0KGgoAAAANSUhEUgAAALQAAAC0CAYAAAAA9zQYyAAAAAXNSR0IArs4c6",
        "appName": "leven-uts-equipment",
        "usedTime": 82
      }
    ]
  },
  "message": ""
}
```

获取应用的使用情况

```
"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	应用列表
data.list.usedCount	Integer	使用次数
data.list.packageName	String	应用包名
data.list.icon	String	应用图标，base64数据(png)
data.list.appName	String	应用名称
data.list.usedTime	String	使用时长，单位：秒
code	Integer	返回类型，0.成功，其他：失败

获取设备安装的app

方法名

getInstallApps

不包含系统应用

用法

- 用法如下：

```
module.getInstallApps(res => {  
  this.writeLog("获取设备安装的app：" + JSON.stringify(res))  
});
```

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "list": [  
      {  
        "versionCode": 1,  
        "packageName": "uni.UNIB291F5B",  
        "lastUpdateTime": 1724598778939,  
        "appName": "leven-uts-equipment",  
        "firstInstallTime": 1724598778939,  
        "versionName": "1.0.0",  
        "icon": "iVBORw0KGgoAAAANSUhEUgAAALQAAAC0CAYAAAAA9zQYyAAAAAXNSR0IArs4c6"  
      }  
    ]  
  },  
  "message": "",
```

获取设备安装的app

```
"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	应用列表
data.list.versionCode	Integer	版本号
data.list.packageName	String	应用包名
data.list.lastUpdateTime	Integer	最后更新时间，单位：毫秒
data.list.appName	String	应用名称
data.list.firstInstallTime	Integer	首次安装时间，单位：毫秒
data.list.versionName	String	版本名称
data.list.icon	String	应用图标，base64数据(png)
code	Integer	返回类型，0.成功，其他：失败

申请文件读写权限

方法名

filePermission

用法

- 用法如下：

```
module.filePermission(res => {
  this.writeLog("申请文件权限：" + JSON.stringify(res))
});
```

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "grantedList": [
      "android.permission.READ_EXTERNAL_STORAGE",
      "android.permission.WRITE_EXTERNAL_STORAGE"
    ]
  },
  "message": "权限被拒绝",
  "code": 7010001
}
```

- 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.grantedList	Array	申请的权限列表，被拒绝的时候会返回

申请文件读写权限

参数名	参数类型	参数描述
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("检查是否有“所有文件”访问权限:" + JSON.stringify(res))  
});
```

- 参数说明

无

回调

- 示例

```
{  
  "data": {  
    "status": false  
  },  
  "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	访问状态，true：有访问权限，false：无访问权限
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("跳转到所有文件访问权限页面:" + JSON.stringify(res))
});
```

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "status": false
  },
  "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	访问状态，true：有访问权限，false：无访问权限
code	Integer	返回类型，0.成功，其他：失败

截图(应用内)

方法名

screenShot

用法

- 用法如下：

```
module.screenShot(res => {
  this.writeLog("截图 ( 应用内 )" + JSON.stringify(res))
});
```

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "path": "/storage/emulated/0/Pictures/LevenUtsEquipment/20240826002724.jpg"
  },
  "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.path	String	图片保存路径
code	Integer	返回类型，0.成功，其他：失败

监听前台通知服务

方法名

setForegroundServiceListener

监听开启和关闭前台通知状态，此方法可以不调用，不影响应用外截图，只是提供开启和关闭的状态通知

用法

- 用法如下：

```
module.setForegroundServiceListener(res => {
  this.writeLog("监听前台服务通知:" + JSON.stringify(res))
});
```

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "type": "success"
  },
  "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	String	服务状态，success：开启，stop：关闭，error：错误

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

取消监听前台通知服务

方法名

clearForegroundServiceListener

用法

- 用法如下：

```
module.clearForegroundServiceListener(res => {
  this.writeLog("取消监听前台通知服务:" + 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

开启前台通知服务

方法名

startForegroundService

应用外截图必须
如果修改通知图标，请将图标文件(png)放置到【uni_modules】>【leven-uts-equipment】>【utssdk】>【app-android】>【res】>【mipmap】目录下

用法

- 用法如下：

```
module.startForegroundService({
  title: "前台通知标题",
  description: "前台通知描述"
}, res => {
  this.writeLog("开启前台通知服务:" + 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	是	无	通知栏标题
description	String	是	无	通知栏描述

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

方法名

stopForegroundService

用法

- 用法如下：

```
module.stopForegroundService(res => {
  this.writeLog("关闭前台通知服务:" + 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

开启截图服务

方法名

openScreenShortService

应用外截图必须

用法

- 用法如下：

```
module.openScreenShortService(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	截图服务是否开启成功

开启截图服务

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

是否开启截图服务

方法名

is0openScreenShortService

用法

- 用法如下：

```
module.is0openScreenShortService(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	截图服务是否开启成功
code	Integer	返回类型，0.成功，其他：失败

截图(应用外)

方法名

screenShotWindow

需开启前台通知服务和截图服务

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "path": "/storage/emulated/0/Pictures/LevenUtsEquipment/20240826005056.jpg"
  },
  "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.path	String	截图后的路径

截图(应用外)

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

屏幕状态监听

方法名

screenListener

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {
    "status": "screenOff"
  },
  "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	String	screenOff：息屏，screenOn：亮屏
code	Integer	返回类型，0.成功，其他：失败

取消屏幕状态监听

方法名

removeScreenListener

用法

- 用法如下：

```
module.removeScreenListener(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

是否注册屏幕状态监听

方法名

isScreenListener

用法

- 用法如下：

```
module.isScreenListener(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	监听状态：true：已监听屏幕状态，false：未监听屏幕状态
code	Integer	返回类型，0.成功，其他：失败

申请悬浮窗权限

方法名

requestAlertWindow

用法

- 用法如下：

```
module.requestAlertWindow(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	悬浮窗开启状态，true：已开启，false：未开启
code	Integer	返回类型，0.成功，其他：失败

是否开启悬浮窗

方法名

isAlertWindow

用法

- 用法如下：

```
module.isAlertWindow(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	悬浮窗开启状态，true：已开启，false：未开启
code	Integer	返回类型，0.成功，其他：失败