

uniapp 安卓保活功能原生插件

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
开启保活服务	2.1
后台运行	2.2
前台显示(v1.1.1已废弃)	2.3
开启悬浮窗	2.4
打开悬浮窗设置页面	2.5
检测是否开启悬浮窗 v1.2.0	2.6
忽略电池优化	2.7
打开电池优化应用设置页面	2.8
检测是否开启忽略电池优化 v1.2.0	2.9
自启动管理	2.10
停止保活 v1.1.0	2.11
更新前台通知 v1.3.0	2.12

使用方法

介绍

安卓保活原生插件，多种技术保活方案大幅提高保活效率，支持多任务app隐藏，息屏保活，清理后继续保活等

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=16644>

示例文件

如果下载的文件不是zip文件，请将扩展名改为zip即可

演示程序

百度网盘

下载地址：<https://pan.baidu.com/s/1uguVT7sMgbFgspeSe0B2ww>

提取码：8jhj

权限

使用方法

- android.permission.REORDER_TASKS
- android.permission.SYSTEM_ALERT_WINDOW
- android.permission.SYSTEM_OVERLAY_WINDOW
- android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
- android.permission.FOREGROUND_SERVICE
- android.permission.RECEIVE_BOOT_COMPLETED
- android.permission.PACKAGE_USAGE_STATS

用法

在需要使用插件的页面加载以下代码

```
const module = uni.requireNativePlugin("leven-alive-AliveModule");
```

页面内容

```
<template>
  <view>
    <uni-card title="安卓保活功能原生插件">
      <button type="primary" @click="openAliveService">开启保活服务</button>
      <button type="primary" @click="backgroundRun">进入后台运行</button>
      <button type="primary" @click="requestOverlay">开启悬浮窗</button>
      <button type="primary" @click="overlaySettings">打开悬浮窗设置页面</button>
      <button type="primary" @click="ignoreBattery">忽略电池优化</button>
      <button type="primary" @click="batterySettings">打开电池优化应用设置页面</button>
      <button type="primary" @click="autoStartSettings">自启动管理</button>
      <button type="primary" @click="stopAliveService">停止保活服务</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-alive-AliveModule");
  export default {
    data() {
```

```

    return {
      logStr: "",
    },
  },
  methods: {
    //开启保活服务
    openAliveService() {
      module.openAliveService({
        //前台通知标题
        title: "保活前台标题",
        //前台通知描述
        description: "保活描述",
        //是否隐藏多任务列表中app
        hideMultitask: false,
        //不息屏是否定时发送服务状态，默认：false
        isScreenOnSendStatus: true,
        // 不息屏发送服务状态时间间隔，单位：秒，默认：30
        screenOnStatusTime: 10,
      }, res => {
        this.writeLog(JSON.stringify(res))
      });
    },
    //后台运行
    backgroundRun() {
      module.backgroundRun(res => {
        this.writeLog(JSON.stringify(res))
      });
    },
    //开启悬浮窗
    requestOverlay() {
      module.requestOverlay(res => {
        this.writeLog(JSON.stringify(res))
      });
    },
    //打开悬浮窗设置页面
    overlaySettings() {
      module.overlaySettings(res => {
        this.writeLog(JSON.stringify(res))
      });
    },
    //忽略电池优化
    ignoreBattery() {
      module.ignoreBattery(res => {
        this.writeLog(JSON.stringify(res))
      });
    },
    //打开电池优化应用设置页面
    batterySettings() {
      module.batterySettings(res => {
        this.writeLog(JSON.stringify(res))
      });
    },
  },
}

```

```
    });  
  },  
  //设置自启动  
  autoStartSettings() {  
    module.autoStartSettings(res => {  
      this.writeLog(JSON.stringify(res))  
    });  
  },  
  //停止保活服务  
  stopAliveService() {  
    module.stopAliveService(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>
```

授权

离线打包授权：云插件价格 * 10

源码授权：云插件价格 * 20

有需要的同学可以联系作者

更新日志

2024-10-08 v1.3.0

- [新增] 新增方法【[更新前台通知](#)】

2024-04-19 v1.2.1

- [优化] 修复安卓13开启保活闪退的bug

2024-04-10 v1.2.0

- [新增] 新增接口【[检测是否开启悬浮窗](#)】
- [新增] 新增接口【[检测是否开启忽略电池优化](#)】

2024-03-28 v1.1.2

- [优化] 开启保活新增参数【是否隐藏通知栏】

2024-03-21 v1.1.1

- [优化] 优化保活功能
- [移除] 移除接口【前台显示】改为点击通知栏自动进入前台

2024-02-23 v1.1.0

- [优化] 优化保活功能
- [新增] 新增接口【[停止保活](#)】

2024-02-06

首次发布

常见问题

如何修改通知栏图标

在项目根目录下创建【nativeplugins】文件夹在文件夹下创建图片文件，路径如下：【leven-alive】> 【android】> 【res】> 【mipmap】> icon.png，(下载文件后将扩展名改为zip即可)

开启保活服务

方法名

openAliveService

用法

- 用法如下：

```
module.openAliveService({
  //前台通知标题
  title: "保活前台标题",
  //前台通知描述
  description: "保活描述",
  //是否隐藏多任务列表中app
  hideMultitask: true,
  //不息屏是否定时发送服务状态，默认：false
  isScreenOnSendStatus: true,
  // 不息屏发送服务状态时间间隔，单位：秒，默认：30
  screenOnStatusTime: 10
}, 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	是	无	前台通知标题
description	String	是	无	前台通知描述
hideMultitask	Boolean	否	false	是否隐藏多任务列表中app
isScreenOnSendStatus	Boolean	否	false	不息屏是否定时发送服务状态
screenOnStatusTime	Boolean	否	30	不息屏发送服务状态时间间隔，单位：秒

参数名	参数类型	是否必填	默认值	参数描述
hideNotification v1.1.2	Boolean	否	false	是否隐藏通知栏， v1.1.3移除

回调

- 示例

```
{
  "data": {
    "status": "startWorking"
  },
  "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	当前服务运行状态，具体请参考下方说明
data.title v1.3.0	String	更新的前台通知标题，更新前台通知时返回
data.description v1.3.0	String	更新的前台通知描述，更新前台通知时返回
code	Integer	返回类型，0.成功，其他：失败

- status状态说明

返回值	说明
startWorking	开始启动服务
workingConnected	服务启动成功
onWorking	服务运行中，在息屏或开启不息屏是否定时发送服务时状态返回

返回值	说明
screenOff	锁屏
screenOn	亮屏
onStop	服务停止， <i>一般不会返回</i>
notificationClick	通知栏点击事件
foreground	前台显示
back	后台运行
notificationChanged v1.3.0	更新前台通知

后台运行

方法名

backgroundRun

用法

- 用法如下：

```
module.backgroundRun(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

前台显示(v1.1.1已废弃)

方法名

foregroundRun v1.1.1已废弃

用法

- 用法如下：

```
module.foregroundRun(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

开启悬浮窗

方法名

requestOverlay

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "未开启悬浮窗",
  "code": -1
}
```

- 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

打开悬浮窗设置页面

方法名

overlaySettings

用法

- 用法如下：

```
module.overlaySettings(res => {
  this.writeLog(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	悬浮窗设置状态，true：悬浮窗开启，false：悬浮窗未开启
code	Integer	返回类型，0.成功，其他：失败

检测是否开启悬浮窗 v1.2.0

方法名

checkOverlay v1.2.0

用法

- 用法如下：

```
module.checkOverlay(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

忽略电池优化

方法名

ignoreBattery

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

```
{
  "data": {},
  "message": "未开启电池优化",
  "code": -1
}
```

- 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

打开电池优化应用设置页面

方法名

batterySettings

用法

- 用法如下：

```
module.batterySettings(res => {
  this.writeLog(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	忽略电池优化状态，true：已忽略电池优化，false：未忽略电池优化
code	Integer	返回类型，0.成功，其他：失败

检测是否开启忽略电池优化 v1.2.0

方法名

checkIgnoreBattery v1.2.0

用法

- 用法如下：

```
module.checkIgnoreBattery(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

自启动管理

方法名

autoStartSettings

用法

- 用法如下：

```
module.autoStartSettings(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

停止保活 v1.1.0

方法名

stopAliveService

用法

- 用法如下：

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

- 参数说明

无

回调

- 示例

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

- 回调说明：

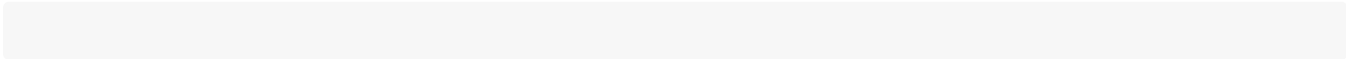
参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
code	Integer	返回类型，0.成功，其他：失败

更新前台通知 v1.3.0

方法名

用法

- 用法如下：

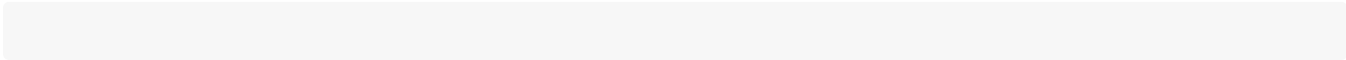


- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
title	String	否	原标题	前台通知标题
description	String	否	原描述	前台通知描述

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
code	Integer	返回类型，0.成功，其他：失败