

uniapp 安卓word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
插件方法	2
申请插件所需权限	2.1
检查是否有“所有文件”访问权限	2.2
跳转到所有文件访问权限页面	2.3
加载文件	2.4
获取页眉和页脚	2.5
获取元数据	2.6
获取标签	2.7
获取表格	2.8
获取段落	2.9
获取小节	2.10
获取图片	2.11
保存图片	2.12
模板替换	2.13
释放资源	2.14

使用方法

介绍

安卓word操作，支持获取页眉、页脚、表格、图片、标签、段落等，支持保存文档图片，文档模版替换等

当前仅支持doc文件操作，不支持docx格式

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=18919>

权限

1. android.permission.WRITE_EXTERNAL_STORAGE
2. 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-word-WordModule");
```

页面内容

```
<template>
  <view>
    <uni-card title="安卓word操作原生插件">
      <button type="primary" @click="requestPermissions">申请插件所需权限</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="loadWord">加载文件</button>
      <button type="primary" @click="getHeadFoot">获取页眉和页脚</button>
      <button type="primary" @click="getWordInfo">获取元数据</button>
      <button type="primary" @click="getBookmarks">获取标签</button>
      <button type="primary" @click="getTables">获取表格</button>
      <button type="primary" @click="getParagraphs">获取段落</button>
      <button type="primary" @click="getSections">获取小节</button>
      <button type="primary" @click="getPictures">获取图片</button>
      <button type="primary" @click="savePictures">保存图片</button>
      <button type="primary" @click="templateReplace">模板替换</button>
      <button type="primary" @click="release">释放资源</button>
      <button type="primary" @click="logStr = ''">清空日志</button>
    </uni-card>
    <uni-card title="图片列表">
      <view class="image-box" v-for="(item, index) in imageList" :key="index">
        <image class="image-body" :src="item"></image>
      </view>
    </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-word-WordModule");
  export default {
    data() {
      return {
```

```

        logStr: "",
        imageList: []
    },
    methods: {
        // 申请插件所需权限
        requestPermissions() {
            module.requestPermissions(res => {
                this.writeLog(JSON.stringify(res))
            });
        },
        // 检查是否有“所有文件”访问权限
        checkAllFilesPermission() {
            module.checkAllFilesPermission(res => {
                this.writeLog(JSON.stringify(res))
            });
        },
        // 跳转到所有文件访问权限页面
        toAllFilesPermissionPage() {
            module.toAllFilesPermissionPage(res => {
                this.writeLog(JSON.stringify(res))
            });
        },
        // 加载文件
        loadWord() {
            module.loadWord({
                //文件路径
                path: "/storage/emulated/0/levenOffice/4.doc"
            }, res => {
                this.writeLog(JSON.stringify(res))
            });
        },
        // 获取页眉和页脚
        getHeadFoot() {
            module.getHeadFoot(res => {
                this.writeLog(JSON.stringify(res))
            });
        },
        // 获取元数据
        getWordInfo() {
            module.getWordInfo(res => {
                this.writeLog(JSON.stringify(res))
            });
        },
        // 获取标签
        getBookmarks() {
            module.getBookmarks(res => {
                this.writeLog(JSON.stringify(res))
            });
        },
    },

```

```

// 获取表格
getTables() {
  module.getTables(res => {
    this.writeLog(JSON.stringify(res))
  });
},
// 获取段落
getParagraphs() {
  module.getParagraphs(res => {
    this.writeLog(JSON.stringify(res))
  });
},
// 获取小节
getSections() {
  module.getSections(res => {
    this.writeLog(JSON.stringify(res))
  });
},
// 获取图片
getPictures() {
  module.getPictures(res => {
    // this.writeLog(JSON.stringify(res))
    if (res.code == 0 && res.data && res.data.list && Array.isArray(res.data.list)) {
      this.imageList = res.data.list.map(item => {
        return "data:image/jpeg;base64," + item.base64;
      })
    }
  });
},
// 保存图片
savePictures() {
  module.savePictures(res => {
    this.writeLog(JSON.stringify(res))
  });
},
// 模板替换
templateReplace() {
  module.templateReplace({
    //替换的模板内容
    list: [{
      key: "${title}",
      content: "天河二号超级计算机"
    }, {
      key: "${generaSituation}",
      content: "由国防科大研制的天河二号超级计算机系统，以峰值计算速度每秒5.49亿亿次、并
    }, {
      key: "${jiedian}",
      content: "16,000"
    }, {
      key: "${cpu}",

```

```

        content: "英特尔Xeon IveBridge E5-2692 ( 2.2GHZ)32,000颗"
      }, {
        key: "{$singleCpu}",
        content: "2"
      }, {
        key: "{$singleCpuCore}",
        content: "12"
      }, {
        key: "{$singleAssistCpu}",
        content: "3"
      }, {
        key: "{$singleAssistCpuCore}",
        content: "57"
      }, {
        key: "{$system}",
        content: "麒麟操作系统 (Kylin Linux)"
      }, {
        key: "{$power}",
        content: "17.6MW ( 整机附带散热系统时为24MW )"
      }, {
        key: "{$componentsSummary}",
        content: "CPU Cores : 384,000\nAccelerator/CP : 48,000"
      }, {
        key: "{$singleNodeStorage}",
        content: "64GB"
      }, {
        key: "{$InternalMemory}",
        content: "1,375TB ( 1,000TB为系统存储器, 375TB为协处理器独占 )"
      }, {
        key: "{$externalMemory}",
        content: "12.4PB"
      }, {
        key: "{$calculationRate}",
        content: "54.9PFLOPS ( 理论峰值 ) \n33.86PFLOPS ( 实际峰值 )"
      }
    ],
    //保存的文件路径(绝对路径),可以为空,如果为空自动保存
    savePath: ""
  }, res => {
    this.writeLog(JSON.stringify(res))
  });
},
// 释放资源
release() {
  module.release(res => {
    this.writeLog(JSON.stringify(res))
  });
},
// 写日志
writeLog(str) {
  console.log(str)
}

```

使用方法

```
        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 scoped>
    .image-box {
        width: 150px;
        height: 150px;
    }
</style>
```

更新日志

2024-06-21

首次发布

申请插件所需权限

- 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

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

加载文件

方法名

loadWord

文档操作所有的接口都需要在加载文件之后调用，当前仅支持doc文件操作，不支持docx格式

用法

- 用法如下：

```
module.loadWord({
  //文件路径
  path: "/storage/emulated/0/levenOffice/4.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	否	无	word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

加载文件

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

获取页眉和页脚

方法名

getHeadFoot

用法

- 用法如下：

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

- 参数说明
无

回调

- 示例

```
{
  "data": {
    "foot": "超级计算机\n",
    "head": "天河二号超级计算机\n"
  },
  "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.foot	String	页脚
data.head	String	页眉
code	Integer	返回类型，0.成功，其他：失败

获取元数据

方法名

getWordInfo

用法

- 用法如下：

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

- 参数说明
无

回调

- 示例

```
{  
  "data": {  
    "lastAuthor": "",  
    "lastSaveTime": 1718848211000,  
    "editTime": 943562880000000000,  
    "createTime": 1531633140000,  
    "author": "",  
    "security": 0,  
    "appName": "WPS Office_12.1.0.16929_F1E327BC-269C-435d-A152-05C5408002CA",  
    "wordCount": 1271,  
    "pageCount": 5,  
    "company": "",  
    "template": "Normal.dotm"  
  },  
  "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.lastAuthor	String	最后编辑人员
data.lastSaveTime	Integer	最后保存时间
data.editTime	Integer	编辑时间
data.createTime	Integer	创建时间
data.author	String	作者
data.security	Integer	安全性
data.appName	String	应用名称
data.wordCount	Integer	文档字数
data.pageCount	Integer	文档页数
data.company	String	公司
data.template	String	模板
code	Integer	返回类型，0.成功，其他：失败

获取标签

- 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.list	Array	标签列表
data.list.name	String	标签名称
data.list.start	Integer	开始位置
data.list.end	Integer	结束位置
code	Integer	返回类型，0.成功，其他：失败

获取表格

方法名

getTables

用法

- 用法如下：

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

- 参数说明
无

回调

- 示例

```
{  
  "data": {  
    "list": [  
      {  
        "list": [  
          {  
            "list": [  
              {  
                "text": "节点数",  
                "column": 0  
              },  
              {  
                "text": "${$jiedian}",  
                "column": 1  
              }  
            ],  
            "row": 0  
          },  
          {  
            "list": [  
              {
```

```
        "text": "中央处理器",
        "column": 0
    },
    {
        "text": "${cpu}",
        "column": 1
    }
],
"row": 1
},
{
    "list": [
        {
            "text": "单节点处理器数",
            "column": 0
        },
        {
            "text": "${singleCpu}",
            "column": 1
        }
    ],
    "row": 2
},
{
    "list": [
        {
            "text": "单处理器核心数",
            "column": 0
        },
        {
            "text": "${singleCpuCore}",
            "column": 1
        }
    ],
    "row": 3
},
{
    "list": [
        {
            "text": "单节点协处理器数",
            "column": 0
        },
        {
            "text": "${singleAssistCpu}",
            "column": 1
        }
    ],
    "row": 4
},
{
```

```
      "list": [  
        {  
          "text": "单协处理器核心数",  
          "column": 0  
        },  
        {  
          "text": "${singleAssistCpuCore}",  
          "column": 1  
        }  
      ],  
      "row": 5  
    },  
    {  
      "list": [  
        {  
          "text": "操作系统",  
          "column": 0  
        },  
        {  
          "text": "${system}",  
          "column": 1  
        }  
      ],  
      "row": 6  
    },  
    {  
      "list": [  
        {  
          "text": "峰值功率",  
          "column": 0  
        },  
        {  
          "text": "${power}",  
          "column": 1  
        }  
      ],  
      "row": 7  
    },  
    {  
      "list": [  
        {  
          "text": "组件汇总",  
          "column": 0  
        },  
        {  
          "text": "${componentsSummary}",  
          "column": 1  
        }  
      ],  
      "row": 8  
    }  
  ]  
}
```

```
},
{
  "list": [
    {
      "text": "单节点存储",
      "column": 0
    },
    {
      "text": "${singleNodeStorage}",
      "column": 1
    }
  ],
  "row": 9
},
{
  "list": [
    {
      "text": "内部存储器",
      "column": 0
    },
    {
      "text": "${InternalMemory}",
      "column": 1
    }
  ],
  "row": 10
},
{
  "list": [
    {
      "text": "外部存储器",
      "column": 0
    },
    {
      "text": "${externalMemory}",
      "column": 1
    }
  ],
  "row": 11
},
{
  "list": [
    {
      "text": "运算速率",
      "column": 0
    },
    {
      "text": "${calculationRate}",
      "column": 1
    }
  ]
}
```

获取表格

```
    ],
    "row": 12
  }
],
"rowNum": 13
}
]
},
"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	表格列表
data.list.rowNum	Integer	行数
data.list.list	Array	行列表
data.list.list.row	Integer	行索引
data.list.list.list	Array	当前行的列列表
data.list.list.list.text	String	当前行的单元格的内容
data.list.list.list.column	Integer	当前行的单元格的索引
code	Integer	返回类型，0.成功，其他：失败

获取段落

方法名

getParagraphs

用法

- 用法如下：

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

- 参数说明
无

回调

- 示例

```
{  
  "data": {  
    "list": [  
      {  
        "text": "【第1张幻灯片】\r"  
      },  
      {  
        "text": "${title}\r"  
      },  
      {  
        "text": "——2014年再登世界超算榜首\r"  
      },  
      {  
        "text": "【第2张幻灯片】\r"  
      },  
      {  
        "text": "一、概况\r"  
      },  
      {  
        "text": "\r"  
      },  
    ]  
  }  
}
```

```
{
  "text": "{$generaSituation}\r"
},
{
  "text": "\u0001\r"
},
{
  "text": "\r"
},
{
  "text": "【第3张幻灯片】\r"
},
{
  "text": "一、概况\r"
},
{
  "text": "\r"
},
{
  "text": "天河二号超级计算机系统由170个机柜组成，包括125个计算机柜、8个服务机柜、1
},
{
  "text": "\r"
},
{
  "text": "天河二号运算1小时，相当于13亿人同时用计算器计算一千年，其存储总容量相当于
},
{
  "text": "\r"
},
{
  "text": "\r"
},
{
  "text": "【第4张幻灯片】\r"
},
{
  "text": "二、特点\r"
},
{
  "text": "\r"
},
{
  "text": "高性能，峰值速度和持续速度都创造了新的世界纪录；\r"
},
{
  "text": "低能耗，能效比为每瓦特19亿次，达到了世界先进水平；\r"
},
{
  "text": "应用广，主打科学工程计算，兼顾了云计算；\r"
```

```
},
{
  "text": "易使用，创新发展了异构融合体系结构，提高了软件兼容性和易编程性；\r"
},
{
  "text": "性价比高，性能世界第一，研制经费仅为美国“泰坦”超级计算机的一半。\r"
},
{
  "text": "\r"
},
{
  "text": "【第5张幻灯片】\r"
},
{
  "text": "三、技术参数\r"
},
{
  "text": "\r"
},
{
  "text": "节点数\u0007"
},
{
  "text": "${jiedian}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "中央处理器\u0007"
},
{
  "text": "${cpu}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "单节点处理器数\u0007"
},
{
  "text": "${singleCpu}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "单处理器核心数\u0007"
},
{
```

```
"text": "{$singleCpuCore}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "单节点协处理器数\u0007"
},
{
  "text": "{$singleAssistCpu}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "单协处理器核心数\u0007"
},
{
  "text": "{$singleAssistCpuCore}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "操作系统\u0007"
},
{
  "text": "{$system}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "峰值功率\u0007"
},
{
  "text": "{$power}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "组件汇总\u0007"
},
{
  "text": "{$componentsSummary}\u0007"
},
{
  "text": "\u0007"
},
},
```

```
{
  "text": "单节点存储\u0007"
},
{
  "text": "{$singleNodeStorage}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "内部存储器\u0007"
},
{
  "text": "{$InternalMemory}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "外部存储器\u0007"
},
{
  "text": "{$externalMemory}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "运算速率\u0007"
},
{
  "text": "{$calculationRate}\u0007"
},
{
  "text": "\u0007"
},
{
  "text": "\r"
},
{
  "text": "\r"
},
{
  "text": "【第6张幻灯片】\r"
},
{
  "text": "四、自主创新\r"
},
{
  "text": "\r"
```

```
},  
{  
  "text": "天河二号自主创新了新型异构多态体系结构，在强化科学与工程计算的同时，可高效支  
},  
{  
  "text": "\r"  
},  
{  
  "text": "【第7张幻灯片】\r"  
},  
{  
  "text": "五、应用领域\r"  
},  
{  
  "text": "\r"  
},  
{  
  "text": "\"天河二号\"由280人历时两年多研制完成，耗资约1亿美元。2013年下半年，它将在  
},  
{  
  "text": "\r"  
},  
{  
  "text": "目前，天河二号已应用于生物医药、新材料、工程设计与仿真分析、天气预报、智慧  
},  
{  
  "text": "\r"  
},  
{  
  "text": "\r"  
},  
{  
  "text": "【第8、9、10张幻灯片】\r"  
},  
{  
  "text": "六、图片欣赏\r"  
},  
{  
  "text": "\r"  
},  
{  
  "text": "\u0001\r"  
},  
{  
  "text": "\u0001\r"  
},  
{  
  "text": "\u0001\r"  
},  
{  
  "text": "\u0001\r"
```

```
        "text": "\r"
      },
      {
        "text": "\r"
      },
      {
        "text": "\r"
      },
      {
        "text": "\u0001\r"
      },
      {
        "text": "\u0001\r"
      },
      {
        "text": "\u0001\r"
      },
      {
        "text": "\u0001\r"
      },
      {
        "text": "\u0001\r"
      }
    ]
  },
  "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	段落列表
data.list.text	String	段落内容
code	Integer	返回类型，0.成功，其他：失败

获取小节

方法名

getParagraphs

用法

- 用法如下：

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

- 参数说明
无

回调

- 示例

```
{  
  "data": {  
    "list": [  
      {  
        "marginTop": 1440,  
        "text": "【第1张幻灯片】\r{title}\r—2014年再登世界超算榜首\r【第2张幻灯片】\  
        "pageHeight": 16838,  
        "marginLeft": 1800,  
        "pageWidth": 11906,  
        "marginBottom": 1440,  
        "marginRight": 1800  
      }  
    ]  
  },  
  "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	小节列表
data.list.marginTop	Integer	小节距离顶部距离
data.list.text	String	小节内容
data.list.pageHeight	Integer	页高度
data.list.marginLeft	Integer	小节距离左侧距离
data.list.pageWidth	Integer	页宽度
data.list.marginBottom	Integer	小节距离底部距离
data.list.marginRight	Integer	小节距离右侧距离
code	Integer	返回类型，0.成功，其他：失败

获取图片

方法名

getPictures

用法

- 用法如下：

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

- 参数说明
无

回调

- 示例

```
{
  "data": {
    "list": [
      {
        "base64": ""
      }
    ]
  },
  "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	图片列表

获取图片

参数名	参数类型	参数描述
data.list.base64	String	图片的base64数据
code	Integer	返回类型，0.成功，其他：失败

保存图片

方法名

savePictures

用法

- 用法如下：

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

- 参数说明
无

回调

- 示例

```
{
  "data": {
    "type": "progress",
    "num": 9,
    "current": 9,
    "path": "/storage/emulated/0/Pictures/LevenWord/1718953344190_9.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

保存图片

参数名	参数类型	参数描述
data.type	String	保存状态：start：开始保存， progress：保存进度， complete：保存完成
data.list.num	Integer	文档所有图片数据
data.list.current	Integer	当做保存的图片索引
data.list.path	String	保存后的文件存储路径
code	Integer	返回类型，0.成功，其他：失败

模板替换

方法名

templateReplace

用法

- 用法如下：

```
module.templateReplace({
  //替换的模板内容
  list: [{
    key: "${title}",
    content: "天河二号超级计算机"
  }, {
    key: "${generaSituation}",
    content: "由国防科大研制的天河二号超级计算机系统，以峰值计算速度每秒5.49亿亿次、持续计
  }, {
    key: "${jiedian}",
    content: "16,000"
  }, {
    key: "${cpu}",
    content: "英特尔Xeon IveBridge E5-2692 ( 2.2GHZ)32,000颗"
  }, {
    key: "${singleCpu}",
    content: "2"
  }, {
    key: "${singleCpuCore}",
    content: "12"
  }, {
    key: "${singleAssistCpu}",
    content: "3"
  }, {
    key: "${singleAssistCpuCore}",
    content: "57"
  }, {
    key: "${system}",
    content: "麒麟操作系统 (Kylin Linux)"
  }, {
    key: "${power}",
    content: "17.6MW （整机附带散热系统时为24MW）"
  }, {
    key: "${componentsSummary}",
    content: "CPU Cores : 384,000\nAccelerator/CP : 48,000"
```

```
    }, {
      key: "${singleNodeStorage}",
      content: "64GB"
    }, {
      key: "${InternalMemory}",
      content: "1,375TB ( 1,000TB为系统存储器, 375TB为协处理器独占 )"
    }, {
      key: "${externalMemory}",
      content: "12.4PB"
    }, {
      key: "${calculationRate}",
      content: "54.9PFLOPS ( 理论峰值 ) \n33.86PFLOPS ( 实际峰值 )"
    }
  ],
  //保存的文件路径(绝对路径), 可以为空, 如果为空自动保存
  savePath: ""
}, 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
list	Array	是	无	要替换的文本数组
list.key	String	是	无	要替换的文本
list.content	String	是	无	替换后的文本
savePath	String	否	无	替换后的文件保存路径, 为空则自动保存

回调

示例

```
{
  "data": {
    "savePath": "/storage/emulated/0/Documents/LevenWord/1718954135299.doc"
  },
  "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.savePath	String	替换后文件的保存路径
code	Integer	返回类型，0.成功，其他：失败

释放资源

方法名

release

用法

- 用法如下：

```
module.release(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