

uniapp 安卓腾讯TBS浏览服务原生插件

目 录

使用说明	1
使用方法	1.1
更新日志	1.2
API	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
选择文件	2.4
初始化	2.5
格式判断	2.6
网络地址文件下载(支持重定向)	2.7
打开文档	2.8
组件	3
用法	3.1
组件属性	3.2
config配置	3.2.1
组件方法	3.3
打开预览	3.3.1

使用方法

介绍

安卓腾讯浏览服务原生插件，无X5内核，腾讯浏览服务（Tencent Browsing Service，以下简称TBS）文档SDK提供稳定、安全、高兼容性的本地文档浏览服务，通过在App内集成SDK的形式，接入后可在客户端内直接打开浏览word、excel、ppt、PDF、txt及dwg、chm、OFD等各类文档格式，支持静态集成SDK，接入简单，满足您和用户在自己的客户端内安全、稳定、流畅地浏览文档的需求，保障数据隐私和浏览体验。具体插件的浏览能力请参考官方文档：<https://cloud.tencent.com/document/product/1645/83890> 本插件接入的sdk版本如下：

腾讯浏览服务文档 SDK 公网版

腾讯浏览服务为接入方提供如下公网版本 SDK，请根据需要下载对应的 SDK：

SDK 类型	SDK 版本	最近更新日期	SDK 大小	SDK 下载	说明	开发者
综合格式 64位			8MB	单击下载	支持 doc、docx、rtf、ppt、pptx、xls、xlsx、xlsm、csv、pdf、txt、epub、chm格式文档	深圳市腾讯计算机系统有限公司
综合格式 64位+32位			10MB	单击下载		
综合格式 + dwg 格式 64位	V1.0.5.6000030	2023.11.09	26MB	单击下载	支持综合格式，支持 dwg 格式	
综合格式 + dwg 格式 64位+32位			41MB	单击下载		

如需支持其他格式可联系作者扩展

SDK常见问题

sdk常见问题请参考官方文档
<https://cloud.tencent.com/document/product/1645/83902>

联系作者

使用方法

关注微信公众号可联系作者



插件地址

<https://ext.dcloud.net.cn/plugin?id=17962>

权限

- android.permission.INTERNET
- android.permission.READ_EXTERNAL_STORAGE
- android.permission.WRITE_EXTERNAL_STORAGE

演示程序

百度网盘

下载地址：https://pan.baidu.com/s/123cOXyRT-i_PV7QmH0_lrQ

提取码：1ce9

api用法

用法

在需要使用插件的页面加载以下代码

```
const module = uni.requireNativePlugin("leven-tbsFile-TbsModule");
```

页面内容

```

<template>
  <view>
    <uni-card title="安卓腾讯浏览服务原生插件">
      <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="selectFile">选择文件</button>
      <button type="primary" @click="init">初始化</button>
      <button type="primary" @click="canOpenFileExt">格式判断</button>
      <button type="primary" @click="downloadUrlFile">网络地址文件下载(支持重定向)</button>
      <button type="primary" @click="openFileReader">打开文档</button>
      <button type="primary" @click="logStr = ''">清空日志</button>
    </uni-card>
    <uni-card title="参数配置">
      <uni-section title="licensekey">
        <uni-easyinput v-model="initParams.licensekey" :maxlength="-1" placeholder="请输入licensekey">
      </uni-section>
      <uni-section title="文件路径">
        <uni-easyinput v-model="params.path" :maxlength="-1" placeholder="请输入文件路径">
      </uni-section>
      <uni-section title="网络地址">
        <uni-easyinput v-model="url" :maxlength="-1" placeholder="请输入网络地址">
      </uni-section>
      <uni-section title="开启长按菜单复制功能, 支持 PDF、DOCX、XLSX 和 TXT">
        <uni-data-checkbox v-model="params.longPressMenu" :localdata="booleanType">
      </uni-section>
      <uni-section title="设置 PPT 打开为翻页模式">
        <uni-data-checkbox v-model="params.pptPageModeDefault" :localdata="booleanType">
      </uni-section>
      <uni-section title="设置为文件流打开模式">
        <uni-data-checkbox v-model="params.streamMode" :localdata="booleanType">
      </uni-section>
      <uni-section title="自动跳转到上次浏览结束位置">
        <uni-data-checkbox v-model="params.gotoLastPos" :localdata="booleanType">
      </uni-section>
      <uni-section title="设置文档视图的背景颜色">
        <uni-easyinput v-model="params.contentBgColor" :maxlength="-1" placeholder="请输入背景颜色">
      </uni-section>
      <uni-section title="顶 bar 的颜色">
        <uni-easyinput v-model="params.barBgColor" :maxlength="-1" placeholder="请输入顶 bar 颜色">
      </uni-section>
      <uni-section title="顶 bar 的高度">
        <uni-easyinput v-model="params.barHeight" :maxlength="-1" placeholder="请输入顶 bar 高度">
      </uni-section>
    </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-tbsFile-TbsModule");
  export default {
    data() {
      return {
        logStr: "",
        //初始化配置
        initParams: {
          //licensekey
          licensekey: ""
        },
        //文件预览配置
        params: {
          //文件路径
          path: "",
          //开启长按菜单复制功能，支持 PDF、DOCX、XLSX 和 TXT，默认：true
          longPressMenu: true,
          //设置 PPT 打开为翻页模式，默认：true
          pptPageModeDefault: true,
          //设置为文件流打开模式，默认：false
          streamMode: false,
          //自动跳转到上次浏览结束位置
          gotoLastPos: true,
          //设置文档视图的背景颜色，可以为空，如"#ffffff"
          contentBgColor: "",
          //设置顶 bar 的颜色，可以为空，如"#ffffff"
          barBgColor: "",
          //顶 bar 的高度，填数字，可以为空
          barHeight: ""
        },
        //网络文档地址
        url: "http://www.yeyuboke.com/svg/1.docx",
        //Boolean选择
        booleanType: [{
          text: "是",
          value: true
        }, {
          text: "否",
          value: false
        }],
      }
    },
    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))
  });
},
// 选择文件
selectFile() {
  module.selectFile({
    //文件类型
    type: "*/*"
  }, res => {
    if (res.code == 0 && res.data && res.data.path) {
      this.params.path = res.data.path
    }
    this.writeLog(JSON.stringify(res))
  });
},
//初始化
init() {
  module.init({
    //文件路径
    licenseKey: this.initParams.licensekey
  }, res => {
    this.writeLog(JSON.stringify(res))
  });
},
// 格式判断
canOpenFileExt() {
  module.canOpenFileExt({
    //文件路径
    path: this.params.path
  }, res => {
    this.writeLog(JSON.stringify(res))
  });
},
//网络地址下载
downloadUrlFile() {

```

```

module.downloadUrlFile({
  //网络文件地址
  url: this.url
}, res => {
  // this.writeLog(JSON.stringify(res))
  if (res.code == 0 && res.data && res.data.type == "success") {
    //下载完成
    this.params.path = res.data.path
  }
});
},
//打开文档
openFileReader() {
  module.openFileReader(this.params, 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>

```

组件用法

用法

在需要使用插件的页面加载以下代码

```

<leven-tbsFile v-if="params.path" ref="refLevenTbsFile" style="flex:1; height: 300px"
  @onError="onError">

```

页面内容

```

<template>
  <view>
    <!-- 组件 -->
    <leven-tbsFile v-if="params.path" ref="refLevenTbsFile" style="flex:1; height
      @onError="onError"></leven-tbsFile>
    <!-- 按钮 -->
    <view style="margin-bottom: 20px;">
      <button type="primary" @click="selectFile">选择文件</button>
      <button type="primary" @click="open">打开预览</button>
      <button type="primary" @click="logStr = ''">清空日志</button>
    </view>
    <uni-card title="参数配置">
      <uni-section title="文件路径">
        <uni-easyinput v-model="params.path" :maxlength="-1" placeholder="请输入文件
      </uni-section>
      <uni-section title="开启长按菜单复制功能，支持 PDF、DOCX、XLSX 和 TXT">
        <uni-data-checkbox v-model="params.longPressMenu" :localdata="booleanType'
      </uni-section>
      <uni-section title="设置 PPT 打开为翻页模式">
        <uni-data-checkbox v-model="params.pptPageModeDefault" :localdata="boolean
      </uni-section>
      <uni-section title="设置为文件流打开模式">
        <uni-data-checkbox v-model="params.streamMode" :localdata="booleanType"><,
      </uni-section>
      <uni-section title="自动跳转到上次浏览结束位置">
        <uni-data-checkbox v-model="params.gotoLastPos" :localdata="booleanType"><,
      </uni-section>
      <uni-section title="设置文档视图的背景颜色">
        <uni-easyinput v-model="params.contentBgColor" :maxlength="-1" placeholder
      </uni-section>
    </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>
  const module = uni.requireNativePlugin("leven-tbsFile-TbsModule");
  export default {
    data() {
      return {
        logStr: "",
        //预览文件配置
        params: {
          //文件路径

```

```

    path: "",
    //开启长按菜单复制功能, 支持 PDF、DOCX、XLSX 和 TXT, 默认: true
    longPressMenu: true,
    //设置 PPT 打开为翻页模式, 默认: true
    pptPageModeDefault: true,
    //设置为文件流打开模式, 默认: false
    streamMode: false,
    //自动跳转到上次浏览结束位置
    gotoLastPos: true,
    //设置文档视图的背景颜色, 可以为空, 如"#ffffff"
    contentBgColor: "",
  },
  //Boolean选择
  booleanType: [{
    text: "是",
    value: true
  }, {
    text: "否",
    value: false
  }],
},
methods: {
  selectFile() {
    module.selectFile({
      //文件类型
      type: "*/*"
    }, res => {
      if (res.code == 0 && res.data && res.data.path) {
        this.params.path = res.data.path
      }
      this.writeLog(JSON.stringify(res))
    });
  },
  open() {
    if (this.$refs.refLevenTbsFile) {
      this.$refs.refLevenTbsFile.open(this.params, res => {
        this.writeLog(JSON.stringify(res))
      });
    }
  },
  // 系统错误
  onError(e) {
    this.writeLog("onError:" + JSON.stringify(e))
  },
  // 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

2024-05-05

首次发布

申请插件所需权限

方法名

requestPermissions

用法

- 用法如下：

```
module.requestPermissions(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

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

选择文件

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
type	String	否	*/*	需要展示的文件格式

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.uriString	String	Uri的原生数据
data.path	String	选择的文件路径
code	Integer	返回类型，0.成功，其他：失败

初始化

方法名

init

用法

- 用法如下：

```
module.init({
  //文件路径
  licenseKey: this.initParams.licensekey
}, 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
licenseKey	String	是	无	腾讯申请的licenseKey

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

初始化

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

格式判断

方法名

canOpenFileExt

该方法是判断文件是否能在TBS中打开

用法

- 用法如下：

```
module.canOpenFileExt({
  //文件路径
  path: this.params.path
}, 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	文件路径

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	检测状态，true：能在TBS中打开，false：不能在TBS中打开
code	Integer	返回类型，0.成功，其他：失败

网络地址文件下载(支持重定向)

方法名

downloadUrlFile

用法

- 用法如下：

```
module.downloadUrlFile({
  //网络文件地址
  url: this.url
}, res => {
  this.writeLog(JSON.stringify(res))
  if (res.code == 0 && res.data && res.data.type == "success") {
    //下载完成
    this.params.path = res.data.path
  }
});
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
url	String	是	无	网络地址

回调

- 示例

```
{
  "data": {
    "url": "http://www.yeyuboke.com/svg/1.docx",
    "type": "start"
  },
  "message": "开始下载",
  "code": 0
}

{
  "data": {
```

```

        "percent": 0.008684945952326092,
        "type": "downloading",
        "url": "http://www.yeyuboke.com/svgga/1.docx",
        "size": 6144,
        "totalSize": 707431
    },
    "message": "下载中",
    "code": 0
}

{
    "data": {
        "url": "http://www.yeyuboke.com/svgga/1.docx",
        "type": "success",
        "path": "/storage/emulated/0/Android/data/test.leven.uniplugin.com/files/1
    },
    "message": "下载完成",
    "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.url	String	当前下载的地址
data.type	String	下载类型，start：开始下载， downloading：下载中， success：下载完成，error：下 载出错
data.percent	Float	当前下载的进度(<=1)
data.size	Integer	已下载的大小，单位：字节
data.totalSize	Integer	文件总大小，单位：字节
data.path	String	下载完成后的文件路径
code	Integer	返回类型，0.成功，其他：失败

打开文档

方法名

openFileReader

用法

- 用法如下：

```
module.openFileReader({
  //文件路径
  path: "",
  //开启长按菜单复制功能，支持 PDF、DOCX、XLSX 和 TXT，默认：true
  longPressMenu: true,
  //设置 PPT 打开为翻页模式，默认：true
  pptPageModeDefault: true,
  //设置为文件流打开模式，默认：false
  streamMode: false,
  //自动跳转到上次浏览结束位置
  gotoLastPos: true,
  //设置文档视图的背景颜色，可以为空，如"#ffffff"
  contentBgColor: "",
  //设置顶 bar 的颜色，可以为空，如"#ffffff"
  barBgColor: "",
  //顶 bar 的高度，填数字，可以为空
  barHeight: ""
}, 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	文件路径
longPressMenu	Boolean	否	true	开启长按菜单复制功能，支持 PDF、DOCX、XLSX 和 TXT，默认：true

参数名	参数类型	是否必填	默认值	参数描述
pptPageModeDefault	Boolean	否	true	设置 PPT 打开为翻页模式，默认：true
streamMode	Boolean	否	false	设置为文件流打开模式，默认：false
gotoLastPos	Boolean	否	true	自动跳转到上次浏览结束位置
contentBgColor	String	否	无	设置文档视图的背景颜色，可以为空,如"#ffffff"
barBgColor	String	否	无	设置顶 bar 的颜色，可以为空,如"#ffffff"
barHeight	Integer	否	无	顶 bar 的高度，填数字，可以为空

回调

- 示例

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

{
  "data": {
    "status": "close"
  },
  "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.openType	String	预览模式，normal：普通模式，stream：流模式
data.status	String	close：关闭预览
code	Integer	返回类型，0.成功，其他：失败

用法

在需要使用插件的页面加载以下代码

```
<leven-tbsFile v-if="params.path" ref="refLevenTbsFile" style="flex:1; height: 300px;"
  @onError="onError">
```

js代码如下：

```
data() {
  return {
    //预览文件配置
    params: {
      //文件路径
      path: "",
      //开启长按菜单复制功能，支持 PDF、DOCX、XLSX 和 TXT，默认：true
      longPressMenu: true,
      //设置 PPT 打开为翻页模式，默认：true
      pptPageModeDefault: true,
      //设置为文件流打开模式，默认：false
      streamMode: false,
      //自动跳转到上次浏览结束位置
      gotoLastPos: true,
      //设置文档视图的背景颜色，可以为空，如"#ffffff"
      contentBgColor: "",
    },
  },
},
```

组件属性

config配置

文件路径

参数名	参数类型	是否必填	默认值	参数描述
path	String	是	无	预览的文件路径

开启长按菜单复制功能

参数名	参数类型	是否必填	默认值	参数描述
longPressMenu	Boolean	否	true	开启长按菜单复制功能

设置 PPT 打开为翻页模式

参数名	参数类型	是否必填	默认值	参数描述
pptPageModeDefault	Boolean	否	true	设置 PPT 打开为翻页模式

设置为文件流打开模式

参数名	参数类型	是否必填	默认值	参数描述
streamMode	Boolean	否	false	设置为文件流打开模式

自动跳转到上次浏览结束位置

参数名	参数类型	是否必填	默认值	参数描述
gotoLastPos	Boolean	否	true	自动跳转到上次浏览结束位置

设置文档视图的背景颜色

config配置

参数名	参数类型	是否必填	默认值	参数描述
contentBgColor	String	否	无	设置文档视图的背景颜色，可以为空,如"#ffffff"

组件方法

打开预览

方法名

open

用法

- 用法如下：

```
module.open({
  //文件路径
  path: "",
  //开启长按菜单复制功能，支持 PDF、DOCX、XLSX 和 TXT，默认：true
  longPressMenu: true,
  //设置 PPT 打开为翻页模式，默认：true
  pptPageModeDefault: true,
  //设置为文件流打开模式，默认：false
  streamMode: false,
  //自动跳转到上次浏览结束位置
  gotoLastPos: true,
  //设置文档视图的背景颜色，可以为空,如"#ffffff"
  contentBgColor: "",
}, 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	文件路径
longPressMenu	Boolean	否	true	开启长按菜单复制功能，支持 PDF、DOCX、XLSX 和 TXT，默认：true
pptPageModeDefault	Boolean	否	true	设置 PPT 打开为翻页模式，默认：true

参数名	参数类型	是否必填	默认值	参数描述
streamMode	Boolean	否	false	设置为文件流打开模式，默认：false
gotoLastPos	Boolean	否	true	自动跳转到上次浏览结束位置
contentBgColor	String	否	无	设置文档视图的背景颜色，可以为空,如"#ffffff"

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