uniapp 安卓讯飞离线语音合成(新版)UTS原生插件

目　录

使用方法

介绍

安卓讯飞离线语音合成(新版)UTS原生插件是根据讯飞AIkit通用XTTS合成 Android SDK 文档开发的插件，离线合成能力能将文字信息转化为声音信息，让应用具备离线语音合成的能力。即让机器像人一样开口说话，给开发者的应用配上“嘴巴“。广泛应用于出行导航服务、阅读听书、智能音箱等行业领域。插件UTS开发，支持uniapp和uniapp x

官方SDK文档

https://www.xfyun.cn/doc/tts/AIkit_offline_tts/Android-SDK.html

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=25613

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-xfAiKitTts"
```

示例代码

```
    <template>
  <view>
    <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 type="primary" @click="toAllFilesPermissionPage">跳转到所有文件访问权限页面
    <button type="primary" @click="init">初始化</button>
    <button type="primary" :disabled="startDisabled" @click="start">开始语音合成</bu
  </view>
</template>

<script>
  import {
    initConfig
  } from '../../utils/sdkConfig';

  // const module = uni.requireNativePlugin("leven-xfSdk-TtsModule");
  import * as module from "@/uni_modules/leven-uts-xfAiKitTts"
  export default {
    data() {
      return {
        startDisabled: true
      }
    },
    onLoad() {

    },
    beforeDestroy() {
      //释放资源
      module.destroy({
        //执行能力，可以为空，为空的话默认为初始化sdk的ability
        ability: initConfig.ability,
      }, res => {
        console.log("释放资源")
      })
    },
    methods: {
      init() {
        module.initSdk({
          appID: initConfig.appID,
          apiKey: initConfig.apiKey,
          apiSecret: initConfig.apiSecret,
          workDir: initConfig.workDir,
          ability: initConfig.ability
        }, res => {
          console.log(res)
          if (res.code == 0) {
```

```
            //注册能力监听回调
            this.registerListener();

            this.startDisabled = false;
          }
        })
      },
      //注册能力监听回调
      registerListener() {
        module.registerListener({
          ability: initConfig.ability
        }, res => {
          console.log(res)
        })
      },
      //开始合成
      start() {
        module.start({
          //必填参数,发音人：xiaoyan:中文 女 晓燕；xiaofeng:中文 男 晓峰；catherine:英文
          vcn: "xiaoyan",
          //必填参数,语种：1:中文, 2:英文, 3:法语, 5:日语, 6:俄语, 9:德语, 15:意大利语, 1
          language: 1,
          //必填参数,文本编码：GBK:GBK编码, UTF-8:UTF-8编码, Unicode:Unicode编码
          textEncoding: "UTF-8",
          //必填参数,合成文本
          text: "科大讯飞成立于1999年，总部位于合肥，是中国人工智能领域领军企业，专注智能语音、
          //语调：最小值:0, 最大值:100
          pitch: 50,
          //音量：最小值:0, 最大值:100
          volume: 50,
          //语速：最小值:0, 最大值:100
          speed: 50,
          //英文发音方式，0:引擎自动判断, 1:按字母发音, 2:按单词发音
          // reg: 0,
          //数字发音方式，0:引擎自动判断, 1:按数字发音, 2:按字符串发音
          // rdn: 0,
          //执行能力，可以为空，为空的话默认为初始化sdk的ability
          ability: initConfig.ability,
          //合成类型，1.播放+合成音频，2.仅播放，3.仅合成音频
          type: 1
        }, res => {
          console.log(res)
        })
      },
      requestPermissions() {
        module.requestPermissions({
          //申请权限列表
          permissions: [
            'android.permission.WRITE_EXTERNAL_STORAGE',
            'android.permission.READ_EXTERNAL_STORAGE',
```

```
            'android.permission.INTERNET'
          ]
        }, res => {
          console.log(res)
        })
      },
      //检查所有文件访问权限
      checkAllFilesPermission() {
        module.checkAllFilesPermission(res => {
          console.log(res)
        })
      },
      //跳转到所有文件访问权限页面
      toAllFilesPermissionPage() {
        module.toAllFilesPermissionPage(res => {
          console.log(res)
        })
      }
    }
  }
</script>

<style>
  .content {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
  }

  .logo {
    height: 200rpx;
    width: 200rpx;
    margin-top: 200rpx;
    margin-left: auto;
    margin-right: auto;
    margin-bottom: 50rpx;
  }

  .text-area {
    display: flex;
    justify-content: center;
  }

  .title {
    font-size: 36rpx;
    color: #8f8f94;
```

```
    }
</style>
```

示例文件

请到插件首页通过 `使用 HBuilderX 导入示例项目`

更新日志

2025-10-28

首次发布

申请插件所需权限

方法名

`requestPermissions`

用法

- 用法如下：

```
module.requestPermissions({
  //申请权限列表
  permissions: [
    'android.permission.WRITE_EXTERNAL_STORAGE',
    'android.permission.READ_EXTERNAL_STORAGE',
    'android.permission.INTERNET'
  ]
}, res => {
  console.log(res)
})
```

- 参数说明

  无

回调

- 示例

```
{
    "data": {
        "type": "onGranted",
        "permissions": [
            "android.permission.WRITE_EXTERNAL_STORAGE",
            "android.permission.READ_EXTERNAL_STORAGE",
            "android.permission.INTERNET"
        ],
        "allGranted": 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.type	String	授权类型，onGranted：同意授权，onDenied：拒绝授权
data.permissions	Array[String]	授权的权限，如果type=onDenied时表示被拒绝的权限
data.allGranted	Boolean	是否全部同意，true：是，false：否
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`

> 对于安卓11以上系统需要调用此方法

用法

- 用法如下：

```
module.checkAllFilesPermission(res => {
  console.log(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	是否有所有文件访问权限，true：是，false：否

检查是否有所有文件访问权限

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

跳转到所有文件访问权限页面

方法名

`toAllFilesPermissionPage`

> 对于安卓11以上系统需要调用此方法

用法

- 用法如下：

```
module.toAllFilesPermissionPage(res => {
  console.log(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	是否有所有文件访问权限，true：是，false：否

跳转到所有文件访问权限页面

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

初始化SDK

方法名

`initSdk`

用法

- 用法如下：

```
module.initSdk({
  appID: initConfig.appID,
  apiKey: initConfig.apiKey,
  apiSecret: initConfig.apiSecret,
  workDir: initConfig.workDir,
  ability: initConfig.ability
}, res => {
  console.log(res)
  if (res.code == 0) {
    uni.navigateTo({
      url: "/pages/index/cmd"
    })
  }
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
appID	String	是	无	讯飞申请的应用id
apiKey	String	是	无	离线引擎托管平台创建应用后，生成的唯一应用标识
apiSecret	String	是	无	离线引擎托管平台创建应用后，生成的唯一应用秘钥

参数名	参数类型	是否必填	默认值	参数描述
workDir	String	是	无	SDK工作目录。默认读取能力资源、写SDK日志在此路径下
ability	String	是	无	传入需要用到的一个或多个能力ID，格式为"xxxx1;xxxx2"，多个能力ID中间用 英文分号 隔开，如果注册的能力非空，SDK将对传入的能力ID授权
customDeviceId	String	否	无	用户自定义设备指纹块，默认为空，设置后会成为设备指纹的一部分，建议设置长度低于256
authType	Integer	否	无	离线授权类型（0或1），0-->(默认)设备级授权（DEVICE）和1-->应用级授权（APP）
authInterval	Integer	否	无	在线授权校验间隔时长，默认为300s，可自定义设置，最短为60s，单位秒
resDir	String	否	无	指定资源读取路径，不设置默认从workDir读取

回调

- 示例

```
{
    "data": {
        "type": "AUTH"
    },
    "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	授权类型
code	Integer	返回类型，0.成功，其他：失败

注册能力监听回调

方法名

`registerListener`

> 识别结果可再回调中判断

用法

- 用法如下：

```
module.registerListener({
  ability: initConfig.ability
}, res => {
  console.log("识别数据：" + JSON.stringify(res))
  if (res.code == 0) {
    //判断监听数据
    let type = res.data.type;
    if (type == "onResult") {
      //识别结果
      let outputData = res.data.outputData;
      let handleID = res.data.handleID;
      if (outputData != null && Array.isArray(outputData)) {
        for (let i = 0; i < outputData.length; i++) {
          let outputDataValue = outputData[i];
          //引擎结果的key
          let key = outputDataValue.key;
          /**
           * key的取值以及含义
           * pgs:progressive格式的结果，即可以实时刷屏
           * htk:带有分词信息的结果，每一个分词结果占一行
           * plain:类比于htk，把一句话结果中的所有分词拼成完整一句，若有后处理，则也含有后
           * vad:语音端点检测结果(需要打开vad功能才会返回)bg:前端点，ed:后端点。单位:帧(
           * readable:json格式的结果。
           */
          //识别结果字节
          let value = outputDataValue.value;
          //识别结果字符串
          let valueString = outputDataValue.valueString;
          console.log("handleID:" + handleID)
          console.log("key:" + outputDataValue.key)
          if (key.indexOf("pgs") >= 0 || key.indexOf("plain") >= 0) {
            console.log("key：" + valueString)
```

```
            }
          }

          if (outputData.length > 0 && outputData[0].status == 2) {
            //识别完成
            console.log("识别完成")
            module.finish(res1 => {})
          }
        }
      } else if (type == "onEvent") {
        //事件回调
        let event = res.data.event;
        let eventData = res.data.eventData;
      } else if (type == "onError") {
        //错误回调
        let err = res.data.err;
        let msg = res.data.msg;
      }
    }
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
ability	String	否	无	执行能力，可以为空，为空的话默认为初始化sdk的ability

回调

- 示例

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

- 回调说明：

注册能力监听回调

参数名	参数类型	参数描述
message	String	消息提示
data	Object	数据对象
data.handleID	Integer	句柄ID
data.type	String	类型，onListener：注册成功
code	Integer	返回类型，0.成功，其他：失败

开始合成

方法名

`start`

用法

- 用法如下：

```
module.start({
      //必填参数,发音人：xiaoyan:中文 女 晓燕；xiaofeng:中文 男 晓峰；catherine:英文 女
      vcn: "xiaoyan",
      //必填参数,语种：1:中文, 2:英文, 3:法语, 5:日语, 6:俄语, 9:德语, 15:意大利语, 16
      language: 1,
      //必填参数,文本编码：GBK:GBK编码, UTF-8:UTF-8编码, Unicode:Unicode编码
      textEncoding: "UTF-8",
      //必填参数,合成文本
      text: "科大讯飞成立于1999年，总部位于合肥，是中国人工智能领域领军企业，专注智能语音、
      //语调：最小值:0, 最大值:100
      pitch: 50,
      //音量：最小值:0, 最大值:100
      volume: 50,
      //语速：最小值:0, 最大值:100
      speed: 50,
      //英文发音方式，0:引擎自动判断, 1:按字母发音, 2:按单词发音
      // reg: 0,
      //数字发音方式，0:引擎自动判断, 1:按数字发音, 2:按字符串发音
      // rdn: 0,
      //执行能力，可以为空，为空的话默认为初始化sdk的ability
      ability: initConfig.ability,
      //合成类型，1.播放+合成音频，2.仅播放，3.仅合成音频
      type: 1
    }, res => {
      console.log(res)
    })
```

- 参数说明

开始合成

参数名	参数类型	是否必填	默认值	参数描述
vcn	String	是	无	必填参数,发音人：xiaoyan:中文 女 晓燕；xiaofeng:中文 男 晓峰；catherine:英文 女 catherine
language	Integer	是	无	必填参数,语种：1:中文, 2:英文, 3:法语, 5:日语, 6:俄语, 9:德语, 15:意大利语, 16:韩语, 23:西班牙语, 48:阿拉伯语, 50:阿拉伯语（Eg）, 12:粤语, 8:印地语, 27:泰语
textEncoding	String	是	无	必填参数,文本编码：GBK编码, UTF-8编码, Unicode编码
text	String	是	无	必填参数,合成文本
type	Integer	否	1	合成类型，1.播放+合成音频，2.仅播放，3.仅合成音频
pitch	Integer	否	无	语调：最小值:0, 最大值:100
volume	Integer	否	无	音量：最小值:0, 最大值:100
speed	Integer	否	无	语速：最小值:0, 最大值:100

开始合成

参数名	参数类型	是否必填	默认值	参数描述
reg	Integer	否	无	英文发音方式，0:引擎自动判断, 1:按字母发音, 2:按单词发音
rdn	Integer	否	无	数字发音方式，0:引擎自动判断, 1:按数字发音, 2:按字符串发音
ability	String	否	无	执行能力，可以为空，为空的话默认为初始化sdk的ability

回调

- 示例

```
{
    "data": {
        "status": "fileFinish",
        "path": "/storage/emulated/0/iflytek/audios/1761644451483.pcm"
    },
    "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	当前状态：playStart：开始播放，fileFinish：生成文件完成，playFinish：播放结束
data.path	String	生成的文件路径，type=fileFinish时返回
code	Integer	返回类型，0.成功，其他：失败

停止合成

方法名

`finish`

用法

- 用法如下：

```
module.finish(res1 => {})
```

- 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

释放资源

方法名

`destroy`

用法

- 用法如下：

```
module.destroy({
//执行能力，可以为空，为空的话默认为初始化sdk的ability
ability: initConfig.ability
}, res => {
console.log("释放资源")
})
```

- 参数说明

参数名	参数类型	是否必填	默认值	参数描述
ability	String	否	无	执行能力，可以为空，为空的话默认为初始化sdk的ability

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