广东移动新网格通app

底座JSAPI说明

V0.10

2025年04月

目录

[广东移动新网格通app 1](#_Toc1739171587)

[底座JSAPI说明 1](#_Toc856943392)

[1 概述 3](#_Toc1622252562)

[1.1 文档说明 3](#_Toc746427222)

[1.2 接入方式 3](#_Toc1750338027)

[1.3 统一参数说明 4](#_Toc1700223183)

[1.3.1 输出格式 4](#_Toc1233629699)

[1.3.2格式说明 4](#_Toc1807222955)

[1.3.3返回编码code说明 5](#_Toc2134985164)

[1.4 修订记录 5](#_Toc391393625)

[2 接口说明 6](#_Toc410244614)

[2.1 基础 6](#_Toc1558720628)

[2.1.1 单点登陆(SSO)(获取临时token) 6](#_Toc264585043)

[2.1.2 第三方登录 7](#_Toc1589668411)

[2.1.3打开app底座的小程序、网格通应用、h5网页 8](#_Toc712931350)

[2.1.4 监听拦截物理返回键（安卓） 9](#_Toc1425932837)

[2.1.5 检测权限 10](#_Toc1883174586)

[2.2 界面 11](#_Toc901277416)

[2.2.1 新建一个浏览器页面打开指定的URL 11](#_Toc1567368421)

[2.2.2 关闭当前浏览器 13](#_Toc1726637645)

[2.3 位置 13](#_Toc652377604)

[2.3.1 取得经纬度 13](#_Toc1606372493)

[2.3.2 持续获取经纬度 15](#_Toc138079767)

[2.4 图片 16](#_Toc1424305209)

[2.4.1 下载图片 16](#_Toc297434554)

[2.4.2 从相册选择相片 17](#_Toc1788102509)

[2.4.3 从文件夹取得相片 20](#_Toc752712645)

[2.5 文件 22](#_Toc15260038)

[1.1.1 打开文件 22](#_Toc924904673)

[1.1.2 下载文件 23](#_Toc1386202125)

[1.1.3 选择本地文件 24](#_Toc1996512219)

[1.1.4 上传本地文件 26](#_Toc948880358)

[1.1.5 打开PDF 27](#_Toc618614284)

[2.6 设备 29](#_Toc1081936061)

[2.5.1 调用系统相机进行拍照 29](#_Toc1355338078)

[2.5.2 调用自定义相机进行拍照 33](#_Toc808033217)

[2.5.3 调用扫一扫 36](#_Toc2075178138)

[2.7 语音识别 37](#_Toc237054439)

[2.7.1 开始语音识别 37](#_Toc591791088)

[2.7.2 结束语音识别 38](#_Toc1236046759)

[2.7.3 监听语音识别结果 39](#_Toc1628561082)

[2.8 录音 40](#_Toc1547024159)

[2.8.1 开始录音录制 40](#_Toc1250526084)

[2.8.2 结束录音 41](#_Toc169440599)

[2.8.3 监听语音识别结果 42](#_Toc224831471)

[2.9 悬浮图标 43](#_Toc1318798024)

[2.9.1 设置悬浮图标 43](#_Toc859668681)

[2.9.2 隐藏悬浮图标 44](#_Toc181544551)

[2.10 横竖屏切换 45](#_Toc1792489917)

[3 附录 46](#_Toc1477434903)

[3.1 小程序路由格式说明 46](#_Toc2042488107)

# 概述

## 文档说明

本文档定义新网格通app底座上的WebView容器加载的H5页面可以使用的JSAPI，该JSAPI用于H5页面调用新网格通底座提供的原生能力。

## 接入方式

app加载网页完成后会注入ado对象，h5页面可以监听adoInit事件，然后使用window.ado对象调用jsapi。

监听adoInit事件代码,

connectAdo (callback) {

if (window.ado) {

callback();

} else {

document.addEventListener('adoInit', function () {

console.log('testado init adoInitSuccess event');

callback();

}, false);

}

}

connectAdo(()=> {

console.log('ado中的方法列表：');

window.ado.xx

});

jsapi调用示例：

let params = {

url: 'https://www.baidu.com',

h5AuthType: '9'，

isKeyFunc: '1'

}

window.ado.openUrl(params) .then((res)=> {  
  let data = res.data  
 }).catch(error => {  
  console.log(error)  
 })

## 统一参数说明

### 1.3.1 输出格式

{

code: 1,

message: "错误信息",

data: 返回值

}

### 1.3.2格式说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| code | Number | 1 | 是 | 返回编码，具体参考返回编码说明 |
| message | String | null | 否 | 返回的错误信息 |
| data | Object/String/Number/Array | null | 否 | 返回的正确数据 |

### 1.3.3返回编码code说明

|  |  |
| --- | --- |
| 1 | 成功 |
| 2 | 关闭页面 |
| 1002 | 当前浏览器不支持此方法 |
| 1003 | 参数错误 |
| 1004 | 方法未实现 |
| 1005 | 调用异常 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修订人** | **修订日期** | **修订内容** |
| 0.1 | 仲海啸 | 2023-11-8 | 定义获取单点登录jsapi、打开底座功能jsapi |
| 0.2 | 仲海啸 | 2023-11-15 | 修改初始化jsapi初始化监听方法 |
| 0.3 | 仲海啸 | 2023-11-23 | 新增一些支持的jsapi |
| 0.4 | 仲海啸 | 2023-12-06 | 修改singleSignOn, 新增thirdLogin |
| 0.6 | 仲海啸 | 2024-01-03 | 修改jsapi接入代码，增加扫码jsapi |
| 0.7 | 仲海啸 | 2024-04-16 | 修改openUrl，增加needResult参数，获取目标小程序的返回值 |
| 0.8 | 刘博瑞 | 2025-02-13 | 增加语音识别能力 |
| 0.9 | 刘博瑞 | 2025-03-21 | 增加录音能力，悬浮图标设置 |
| 0.10 | 刘博瑞 | 2025-04-01 | 增加横竖屏切换 |

# 接口说明

## 基础

### 2.1.1 单点登陆(SSO)(获取临时token)

#### 方法名称

singleSignOn

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| appId | String | 无 | 是 | 平台分配的应用标识符（需要找找刘龙汉30001736或平台的后台开发分配） |

#### 返回值data说明

{

"code": 1,

"message": "success",

"data": {

"token": "eyJRlc3QiLCJjb250ZW50IjoiYlFNYZFVhOHifQ=="

}

}

#### 示例

window.ado.singleSignOn({

appId: ‘400000’

}).then((res) => {

let data = res.data

if(data) {

let token = data.token

}

}).catch(function (err) {

console.log(err)

})

### 2.1.2 第三方登录

#### 方法名称

thirdLogin

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| appId | String | 无 | 是 | 平台分配的应用标识符（需要找找刘龙汉30001736或平台的后台开发分配） |
| ssoTicket | String | 无 | 是 | 第三方SSO凭据（第三方accessToken） |

#### 返回值data说明

{

"code": 1,

"message": "success",

"data": null

}

#### 示例

let params = {

appId: '平台分配的应用标识符',

ssoTicket: '第三方accessToken'

}

window.ado.thirdLogin(params).then((res) => {

console.log(res)

}).catch(function (err) {

console.log(err)

})

### 2.1.3打开app底座的小程序、网格通应用、h5网页

#### 方法名称

openUrl

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| url | String | 无 | 是 | 需要打开的URL地址 1. 当h5AuthType为9时，填目标h5的url   1. 当h5AuthType为8时，填网格通应用链接 2. 当h5AuthType为0时，填小程序跳转路由 |
| h5AuthType | String | 9 | 是 | url类型:  9: app自带浏览器  8: 网格通应用 0: 小程序 |
| isKeyFunc | String | 0 | 是 | 1: 是关键业务，会校验是否需要刷脸刷证  0: 不是关键性业务 |

#### 返回值data说明

返回值是调用方法是否成功的信息

{

"code": 1,

"message": "success",

"data": null

}

#### 示例

let params = {

url: 'https://www.baidu.com',

h5AuthType: '9'，

isKeyFunc: '1'

}

window.ado.openUrl(params) .then((res)=> {  
  let data = res.data  
 }).catch(error => {  
  console.log(error)  
 })

### 2.1.4 监听拦截物理返回键（安卓）

#### 方法名称

backEventListener

#### 适用范围

tabbar浏览器，工作台项浏览器，页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| isInterceptBackPress | Boolean | true | 否 | 是否拦截物理返回键 |

#### 返回值code说明

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| code | Number | 响应编码  1018：已拦截物理返回键  1019：未拦截物理返回键 |

#### 返回值data说明

返回值是调用方法是否成功的回调信息

{

"code": 1018,

"message": "success",

"data": null

}

#### 示例

let isInterceptBackPress = true

window.ado.backEventListener(isInterceptBackPress).then(function (res) {

console.log('点击了物理返回键')

// 监听物理返回键，进行对应操作

// todo

}).catch(function (err) {

console.log(err)

})

### 2.1.5 检测权限

#### 方法名称

checkPermission

#### 适用范围

tabbar浏览器，工作台项浏览器，页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| id | String | 无 | 是 | 唯一id（按id存储权限说明） |
| permissionCode | String | 无 | 是 | 权限编码：  CALENDAR 日历  CAMERA 相机  CONTACTS 通讯录  LOCATION 位置服务  MICROPHONE 麦克风  STORAGE 相册 |
| permissionDescription | String | 无 | 是 | 权限说明 |

#### 返回值data说明

{

code: 1

message: 出错内容，此内容仅用于记录日志

data: true（有权）/ false（无权

checkOnly: false // true: 已经授权，false： 弹出了授权弹框

}

#### 示例

let params = {

id: 'getLocation',

permissionCode: 'LOCATION',

permissionDescription: '需要使用位置权限获取位置信息'

}

window.ado.checkPermission(params).then((res) => {

let data = res.data

}).catch(error => {

console.log(error)

})

## 界面

### 2.2.1 新建一个浏览器页面打开指定的URL

#### 方法名称

open

#### 适用范围

tabbar浏览器，工作台项浏览器，页面浏览器

#### 参数说明

**params [Object]** (详情见7.1.5示例3)

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| url | String | 无 | 是 | 需要打开的URL地址 |
| showTitle | Boolean | false | 否 | 是否显示标题 |
| title | String | 无 | 否 | 标题名称 |
| orientation | String | 无 | 否 | 手机屏幕方法（可选值：auto（自动适配）, horizontal（横屏）, portrait（竖屏）） |
| header | Object | 无 | 否 | 需要传入的请求头参数 |

#### 返回值data说明

返回值是调用方法是否成功的信息

{

"code": 1,

"message": "success",

"data": null

}

#### 示例

**// 示例1**

let url = 'https://www.baidu.com'

let showTitle = true

let title = '打开页面的标题'

let orientation = 'portrait'

window.ado.open (url, showTitle, title, orientation)

**// 示例3**

let params = {

url: 'https://www.baidu.com',

showTitle: true,

title: '打开页面的标题',

orientation: 'portrait'

header: {

headerKey1: '我要传的请求头值1',

headerKey2: '我要传的请求头值2',

…

}

}

window.ado.open (params)

### 2.2.2 关闭当前浏览器

#### 方法名称

close

#### 适用范围

工作台项浏览器，页面浏览器

#### 参数说明

无

#### 返回值data说明

返回值是调用方法是否成功的信息

{

"code": 1,

"message": "success",

"data": null

}

#### 示例

window.ado.close()

## 位置

### 取得经纬度

#### 方法名称

getLatLng

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| timeout | Number | 3000 | 否 | 超时时间 |
| permissionId | String | 无 | 否 | 唯一id（按id存储权限说明） |
| permissionDescription | String | 无 | 否 | 权限说明 |

#### 返回值data说明

{

code: 1(成功)/9001(未开启权限)/9002(未开启定位)/9003(未取到定位数据)

message: 出错内容，此内容仅用于记录日志

data: {

// pre 前一次数据

pre: {

lat: 22.960679758736962,

lng: 113.88831695369365,

time: 1635925700435 //时间戳

},

// cur当前数据

cur: {

lat: 22.960679758736962,

lng: 113.88831695369365,

time: 1635925700435 //时间戳

}

}

}

#### 示例

let params = {

timeout: 3000,

loop: false,

permissionId: 'getLatLng \_1',

permissionDescription: '因xx功能，需要获取定位权限'

}

window.ado.getLatLng(params) .then((res)=> {  
  let data = res.data  
 }).catch(error => {  
  console.log(error)  
 })

### 持续获取经纬度

#### 方法名称

getLatLngLoop

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| timeout | Number | 3000 | 否 | 超时时间 |
| permissionId | String | 无 | 否 | 唯一id（按id存储权限说明） |
| permissionDescription | String | 无 | 否 | 权限说明 |

#### 返回值data说明

{

code: 1(成功)/9001(未开启权限)/9002(未开启定位)/9003(未取到定位数据)

message: 出错内容，此内容仅用于记录日志

data: {

// pre 前一次数据

pre: {

lat: 22.960679758736962,

lng: 113.88831695369365,

time: 1635925700435 //时间戳

},

// cur当前数据

cur: {

lat: 22.960679758736962,

lng: 113.88831695369365,

time: 1635925700435 //时间戳

}

}

}

#### 示例

let params = {

timeout: 3000,

loop: false,

permissionId: 'getLatLng \_1',

permissionDescription: '因xx功能，需要获取定位权限'

}

window.ado.getLatLng(params) .then((res)=> {  
  let data = res.data  
 }).catch(error => {  
  console.log(error)  
 })

## 图片

### 2.4.1 下载图片

#### 方法名称

downImage

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| url | String | 无 | 二选一必填 | 要下载图片的URL |
| base64 | String | 无 | 二选一必填 | 要保存图片的base64字符串 |
| permissionId | String | 无 | 否 | 唯一id（按id存储权限说明） |
| permissionDescription | String | 无 | 否 | 权限说明 |

#### 返回值data说明

返回值是相册中图片的地址路径

{

"code": 1,

"message": "success",

"data": '图片路径'

}

#### 示例

let params = {

url: '要下载图片的url',

permissionId: 'downImage \_1',

permissionDescription: '因xx功能，需要获取文件存储权限'

}

window.ado.downImage(params).then((res) => {

 let data = res.data

}).catch(error => {

  console.log(error);

})

### 2.4.2 从相册选择相片

#### 方法名称

selectImageFromAlbum

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| maxNumber | Int | 1 | 否 | 选择相片的最大数量(正整数，1到10之间) |
| maxWidth | Int | 1000 | 是 | 等比压缩的最大宽度(正整数，取值范围为1至1999) |
| maxHeight | Int | 1000 | 是 | 等比压缩的最大高度(正整数，取值范围为1至1999) |
| quality | Int | 85 | 否 | 压缩质量比(正整数，1到100之间) |
| callbackType | Int | 2 | 否 | 回传类型( 2: Base64文件编码; 3: 本地文件路径 |
| permissionId | String | 无 | 否 | 唯一id（按id存储权限说明） |
| permissionDescription | String | 无 | 否 | 权限说明 |
| cut | Boolean | false | 否 | 是否裁剪图片 |

#### 返回值data说明

1、callbackType=2时，返回值data是一个对象数组，每个对象包含下列参数：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| size | String | 相片的大小 |
| base64 | String | 文件base64字符串 |
| fileName | String | 文件名称 |
| extension | String | 文件扩展名 |

{

"code": 1,

"message": "success",

"data": [{

size: '相片大小',

base64: '文件base64字符串',

fileName: '文件名称',

extension: '文件扩展名'

}]

}

2、callbackType=3时，返回值data是一个对象数组，每个对象包含下列参数：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| size | String | 相片的大小 |
| filePath | String | 文件本地路径 |
| fileName | String | 文件名称 |
| extension | String | 文件扩展名 |

{

"code": 1,

"message": "success",

"data": [{

size: '相片大小',

filePath: '文件本地路径',

fileName: '文件名称',

extension: '文件扩展名'

}]

}

#### 示例

**// 示例1**

let params = {

maxNumber: 1,

maxWidth: 1000,

maxHeight: 1000,

quality: 85,

showProgress: true,

callbackType: 2,

appId: '',

requestUrl: '',

requestHeaders: {},

permissionId: 'selectImageFromAlbum\_1',

permissionDescription: '因xx功能，需要获取相册权限',

cut: false

}

window.ado.selectImageFromAlbum(params).then((res)=> {

let data = res.data

}).catch(error => {

  onsole.log(error)

})

**// 示例2**

let params = {

maxWidth: 1000,

maxHeight: 1000,

callbackType: 3,

permissionId: 'selectImageFromAlbum\_1',

permissionDescription: '因xx功能，需要获取相册权限'

}

window.ado.selectImageFromAlbum(params).then((res)=> {

let data = res.data

}).catch(error => {

  onsole.log(error)

})

### 2.4.3 从文件夹取得相片

#### 方法名称

getImage

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| folderName | String | 无 | 是 | 文件夹名称 |
| imageName | String | 无 | 是 | 相片名称(支持模糊搜索) |
| width | Number | 无 | 否 | 略缩图的宽（单位：像素） |
| height | Number | 无 | 否 | 略缩图的高（单位：像素） |

#### 返回值code说明

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| code | Number | 响应编码  1：调用成功  1021：调用失败，不存在该文件夹名称  1022：调用失败，不存在该相片名称 |

#### 返回值data说明

数组 Array

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| path | String | 相片路径 |
| base64 | String | base64略缩图（略缩图的宽高为空时返回空） |

{

code: 1,

message: "success",

data: [

{

path: '相片地址',

base64: 'base64缩略图字符串'

},

{

path: '相片地址',

base64: 'base64缩略图字符串'

},

…

]

}

#### 示例

let params = {

folderName: '相册文件夹名称',

imageName: '图片名称',

width: 50,

height: 50

}

window.ado.getImage(params).then((res) => {

  let data = res.data

}).catch(error => {

  console.log(error)

})

## 文件

### 打开文件

#### 方法名称

systemOpenFile

#### 适用范围

tabbar浏览器，工作台项浏览器，页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| url | String | 无 | 是 | 要打开文件的URL  注意：一个URL对应一个文件，原生根据URL缓存文件 |

#### 返回值data说明

{

"code": 1,

// 下载文件的存储路径

"data": "/storage/emulated/0/Android/data/com.huawei.nis.android.gridbee.vt3/cache/bc5dc8c4-841f-4624-9af6-253bede3aa8a?appId=49a8119e-8e1f-47c8-9b05-585f11e81b36",

"message": ""

}

#### 示例

let url = '要打开的文件路径'

window.ado.systemOpenFile(url).then((res)=> {

 console.log(res)

 }).catch(error => {

 console.log(error);

 });

### 下载文件

#### 方法名称

downFile

#### 适用范围

tabbar浏览器，工作台项浏览器，页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| url | String | 无 | 是 | 要下载文件的URL  注意：一个URL对应一个文件，原生根据URL缓存文件 |
| config | Object | 无 | 否 | 请求配置 |

**config [Object]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 默认值 | 必填 | 说明 |
| requestHeader | Object | 无 | 否 | 请求头 |
| requestBody | Object | 无 | 否 | 请求体 |
| requestMethod | String | 无 | 否 | 请求方式：get、post等 |

特殊说明：使用requestHeader、requestBody、requestMethod暂不支持非同域的https请求

#### 返回值data说明

{

"code": 1,

// 下载文件的存储路径

"data": "/storage/emulated/0/Android/data/com.huawei.nis.android.gridbee.vt3/cache/bc5dc8c4-841f-4624-9af6-253bede3aa8a?appId=49a8119e-8e1f-47c8-9b05-585f11e81b36",

"message": ""

}

#### 示例

let url = 'http://a/b/c.txt'

let config = {

requestHeader: {

headerKey1: '我要传的请求头值1',

headerKey2: '我要传的请求头值2',

…

},

requestBody: {

bodyKey1: '我要传的请求体值1',

bodyKey2: '我要传的请求体值2',

…

},

requestMethod: 'post'

}

window.ado.downFile(url, config).then((res) => {

 let data = res.data

}).catch(error => {

 console.log(error);

})

### 选择本地文件

#### 方法名称

selectLocalFiles

#### 适用范围

tabbar浏览器，工作台项浏览器，页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| fileNumber | Number | 1 | 否 | 文件数量 |
| type | Array | 无 | 否 | 指定过滤的文件类型选择数组（type不传或为空数组时，不过滤文件类型） |

#### 返回值data说明

返回值是文件数组对象

**Array [Object]**

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| filePath | String | 文件路径 |

{

"code": 1,

"message": "success",

"data": [

{

filePath: '/root/file/介绍文档.docx'

}

]

}

#### 示例

let object = {

fileNumber : 1,

type: ['docx','xlsx', 'jpg', 'png'],

selectType: 'file'

}

window.ado.selectLocalFiles(object).then(function (res) {

let data = res.data

}).catch(function (err) {

console.log(err)

})

### 上传本地文件

#### 方法名称

uploadLocalFiles

#### 适用范围

tabbar浏览器，工作台项浏览器，页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| filePathList | Array | 无 | 是 | 文件路径数组 |
| appId | Sting | 无 | 否 | 回传类型为文件时, 上传到的内置服务器的文件appId。  注意：上传到内置服务器时必填。 |
| requestUrl | Sting | 无 | 否 | 回传类型为文件时,可将文件上传到指定的服务器，不指定则上传到系统内置服务器  注意：上传到指定服务器时必填。 |
| requestHeaders | Object | 无 | 否 | 回传类型为文件时,可将文件上传到指定的服务器，请求header所带的参数 |
| requestParams | Object | 无 | 否 | 回传类型为文件时,可将文件上传到指定的服务器，请求的参数 |

#### 返回值data说明

**2.传入requestUrl，上传至指定服务器时：**

返回值是上传成功后的文件对象

**Object**

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| 文件本地路径 | Object | 对应文件上传接口返回的数据, |

{

"code": 1,

"message": "success",

"data": {

'/root/file/介绍文档.docx': 对应文件上传接口返回的数据,

'/root/file/演讲文档.docx': 对应文件上传接口返回的数据,

}

}

#### 示例

**//示例2：传入reqeustUrl，上传至指定服务器**

let object = {

filePathList : [

'../files/测试文档1.docx',

'../files/测试文档2.docx',

…

],

requestUrl: '上传指定的url',

requestHeaders: {} // 请求头

requestParams: {} // 请求参数

}

window.ado.uploadLocalFiles(params).then(function (res) {

console.log(res.data)

}).catch(function (err) {

console.log(err)

})

### 打开PDF

#### 方法名称

openPDF

#### 适用范围

tabbar浏览器，工作台项浏览器，页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| filePath | String | 无 | 是 | PDF文件路径（支持服务路径和本地路径） |
| title | String | 无 | 否 | 页面标题 |
| config | Object | 无 | 否 | 请求配置 |

**config Object**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 默认值 | 必填 | 说明 |
| requestHeader | Object | 无 | 否 | 请求头 |
| requestBody | Object | 无 | 否 | 请求体 |
| requestMethod | String | 无 | 否 | 请求方式：get、post等 |

特殊说明：使用requestHeader、requestBody、requestMethod暂不支持非同域的https请求

#### 返回值data说明

返回值是调用方法是否成功的回调信息

{

"code": 1,

"message": "success",

"data": null

}

#### 示例

**//示例1：不使用请求配置**

let params = {

filePath: 'http://abc/download/file/previewFile.pdf',

title: '页面标题',

}

window.ado.openPDF(params).then(function (res) {

console.log(res)

}).catch(function (err) {

console.log(err)

})

**//示例2：使用请求配置**

let object = {

filePath: 'http://abc/download/file/previewFile.pdf',

title: '页面标题',

config: {

requestHeader: {},

requestBody: {},

requestMethod: 'post'

}

}

window.ado.openPDF(params).then(function (res) {

console.log(res)

}).catch(function (err) {

console.log(err)

})

## 设备

### 2.5.1 调用系统相机进行拍照

#### 方法名称

selectImageFromSystemCamera

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| maxWidth | Number | 1000 | 是 | 等比压缩的最大宽度(正整数，取值范围为1至1999) |
| maxHeight | Number | 1000 | 是 | 等比压缩的最大高度(正整数，取值范围为1至1999) |
| quality | Number | 85 | 否 | 要下载文件压缩质量比(正整数，1到100之间)的URL |
| callbackType | Number | 1 | 否 | 回传类型(1：文件ID; 2: Base64文件编码;3: 本地文件路径(C15版本支持)) |
| appId | Sting | 无 | 否 | 回传类型为文件时, 上传到的内置服务器的文件appId |
| requestUrl | Sting | 无 | 否 | 回传类型为文件时,可将文件上传到指定的服务器，不指定则上传到系统内置服务器 |
| requestHeaders | Object | 无 | 否 | 回传类型为文件时,可将文件上传到指定的服务器，请求header所带的参数 |
| permissionId | String | 无 | 否 | 唯一id（按id存储权限说明） |
| permissionDescription | String | 无 | 否 | 权限说明 |
| cut | Boolean | false | 否 | 是否裁剪图片 |

#### 返回值data说明

1、callbackType=2时，返回值data是一个对象数组，每个对象包含下列参数

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| size | String | 相片的大小 |
| base64 | String | 文件base64字符串 |
| fileName | String | 文件名称 |
| extension | String | 文件扩展名 |

{

"code": 1,

"message": "success",

"data": [{

size: '相片大小',

base64: '文件base64字符串',

fileName: '文件名称',

extension: '文件扩展名'

}]

}

2、callbackType=3时，返回值data是一个对象数组，每个对象包含下列参数：(C15版本支持)：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| size | String | 相片的大小 |
| filePath | String | 文件本地路径 |
| fileName | String | 文件名称 |
| extension | String | 文件扩展名 |

{

"code": 1,

"message": "success",

"data": [{

size: '相片大小',

filePath: '文件本地路径',

fileName: '文件名称',

extension: '文件扩展名'

}]

}

#### 示例

**// 示例1**

let params = {

maxNumber: 1,

maxWidth: 1000,

maxHeight: 1000,

quality: 85,

showProgress: true,

callbackType: 2,

appId: '',

requestUrl: '',

requestHeaders: {},

permissionId: 'selectImageFromSystemCamera\_1',

permissionDescription: '因xx功能，需要获取相机权限',

cut: false

}

window.ado.selectImageFromSystemCamera(params).then((res)=> {

let data = res.data

}).catch(error => {

  onsole.log(error)

})

**// 示例2**

let params = {

maxNumber: 1,

maxWidth: 1000,

maxHeight: 1000,

quality: 85,

showProgress: true,

callbackType: 2,

permissionId: 'selectImageFromSystemCamera\_2',

permissionDescription: '因xx功能，需要获取相机权限',

cut: false

}

window.ado.selectImageFromSystemCamera(params).then((res)=> {

let data = res.data

}).catch(error => {

  onsole.log(error)

})

**// 示例3**

let params = {

maxWidth: 1000,

maxHeight: 1000,

callbackType: 3,

permissionId: 'selectImageFromSystemCamera \_3',

permissionDescription: '因xx功能，需要获取相机权限'

}

window.ado.selectImageFromSystemCamera(params).then((res)=> {

let data = res.data

}).catch(error => {

  onsole.log(error)

})

### 2.5.2 调用自定义相机进行拍照

#### 方法名称

selectImageFromCustomCamera

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| watermarkText | Array | [] | 否 | 水印文字 |
| watermarkPosition | Number | 1 | 否 | 水印位置(1:左上角; 2: 右上角; 3: 左下角; 4: 右下角) |
| maxWidth | Number | 0 | 否 | 等比压缩的最大宽度(正整数，为0就表示不等比缩放) |
| maxHeight | Number | 0 | 否 | 等比压缩的最大高度(正整数，为0就表示不等比缩放) |
| quality | Number | 85 | 否 | 要下载文件压缩质量比(正整数，1到100之间) |
| showProgress | Boolean | true | 否 | 是否显示进度条(true/false) |
| callbackType | Number | 1 | 否 | 回传类型(1：文件ID; 2: Base64文件编码); 3: 文件id |
| appId | Sting | 无 | 否 | 回传类型为文件时, 上传到的内置服务器的文件appId |
| requestUrl | Sting | 无 | 否 | 回传类型为文件时,可将文件上传到指定的服务器，不指定则上传到系统内置服务器 |
| requestHeaders | Object | 无 | 否 | 回传类型为文件时,可将文件上传到指定的服务器，请求header所带的参数 |
| permissionId | String | 无 | 否 | 唯一id（按id存储权限说明） |
| permissionDescription | String | 无 | 否 | 权限说明 |
| cut | Boolean | false | 否 | 是否裁剪图片 |

#### 返回值data说明

1、callbackType=2时，返回值data是一个对象数组，每个对象包含下列参数

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| size | String | 相片的大小 |
| base64 | String | 文件base64字符串 |
| fileName | String | 文件名称 |
| extension | String | 文件扩展名 |

{

"code": 1,

"message": "success",

"data": [{

size: '相片大小',

base64: '文件base64字符串',

fileName: '文件名称',

extension: '文件扩展名'

}]

}

2、callbackType=3时，返回值data是一个对象数组，每个对象包含下列参数：(C15版本支持)：

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| size | String | 相片的大小 |
| filePath | String | 文件本地路径 |
| fileName | String | 文件名称 |
| extension | String | 文件扩展名 |

{

"code": 1,

"message": "success",

"data": [{

size: '相片大小',

filePath: '文件本地路径',

fileName: '文件名称',

extension: '文件扩展名'

}]

}

#### 示例

**// 示例1**

let params = {

maxNumber: 1,

maxWidth: 1000,

maxHeight: 1000,

quality: 85,

showProgress: true,

callbackType: 2,

appId: '',

requestUrl: '',

requestHeaders: {},

permissionId: 'selectImageFromSystemCamera\_1',

permissionDescription: '因xx功能，需要获取相机权限',

cut: false

}

window.ado.selectImageFromSystemCamera(params).then((res)=> {

let data = res.data

}).catch(error => {

  onsole.log(error)

})

**// 示例2**

let params = {

maxNumber: 1,

maxWidth: 1000,

maxHeight: 1000,

quality: 85,

showProgress: true,

callbackType: 2,

permissionId: 'selectImageFromSystemCamera\_2',

permissionDescription: '因xx功能，需要获取相机权限',

cut: false

}

window.ado.selectImageFromSystemCamera(params).then((res)=> {

let data = res.data

}).catch(error => {

  onsole.log(error)

})

**// 示例3**

let params = {

maxWidth: 1000,

maxHeight: 1000,

callbackType: 3,

permissionId: 'selectImageFromSystemCamera \_3',

permissionDescription: '因xx功能，需要获取相机权限'

}

window.ado.selectImageFromSystemCamera(params).then((res)=> {

let data = res.data

}).catch(error => {

  onsole.log(error)

})

### 2.5.3 调用扫一扫

#### 方法名称

scanCode

#### 适用范围

页面浏览器

#### 参数说明

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| permissionId | String | 无 | 否 | 唯一id（按id存储权限说明） |
| permissionDescription | String | 无 | 否 | 权限说明 |

#### 返回值data说明

#### 返回值是扫码内容

{

    "code": 1,

    "message": "success",

  "data":  扫描内容

}

示例

let protocol = "http, https, friend, invite-code, web-login"

let enabledProtocol = true

let permissionId = 'scanCode\_1',

let permissionDescription = '因xx功能，需要获取相机权限'

window.ado.scanCode({

permissionId,

permissionDescription

}).then((res) => {  
 let data = res.data

}).catch(error => {  
 console.log(error)  
 })

## 语音识别

### 开始语音识别

#### 方法名称

startVoiceRecord

#### 适用范围

页面浏览器

#### 参数说明：

#### 参数说明：

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| pkOn | String | 0 | 否 | 服务层声纹识别开关，打开pk\_on开关：0:关闭，1:打开  默认值：0 |
| tag | String | group\_default | 否 | 内部使用的模型角色名（服务自定义的参数）group\_xinan (西南方言标签,group\_jinrong(金融领域标签), group\_default(默认模型) |
| hotword | String |  | 否 | 用于需要加载临时热词资源的场景，把热词通过该参数传递，仅对本次会话生效  例如：hotword=热词A+热词B+热词B |
| requestPermission | String | 无 | 否 | 是否申请权限 |

**返回值data说明：**

返回值是扫码内容

{

    "code": 1,

    "message": "success",

}

示例

window.ado.startVoiceRecord().then((res) => {

console.log('startVoiceRecord res = ' + JSON.stringify(res))

// if (res.code === 0 && res.resultType === 'message') {

// this.voiceResult = res.result

// }

}).catch(error => {

console.log(error)

})

### 结束语音识别

#### 方法名称

stopVoiceRecord

#### 适用范围

页面浏览器

#### 参数说明：无

**返回值data说明：无**

{

    "code": 1,

    "message": "success",

}

示例

window.ado.stopVoiceRecord().then((res) => {

console.log('stopVoiceRecord')

}).catch(error => {

console.log(error)

})

### 监听语音识别结果

connectAdo (callback) {

if (window.ado) {

callback()

} else {

document.addEventListener('adoInit', function () {

console.log('testado init adoInitSuccess event')

callback()

}, false)

}

},

// 监听语音识别结果事件

mounted () {

this.connectAdo(() => {

window.ado.subscribe('voiceRecordEvent', (res) => {

console.log('handle voiceRecordEvent ', JSON.stringify(res))

if (res.type === 'message') {

this.voiceResult = res.message

}

})

})

},

// 取消监听

unMounted () {

this.connectAdo(() => {

window.ado.unSubscribe('voiceRecordEvent')

})

},

#### 事件返回值voiceRecordEvent说明

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| type | String | start：录音开始  stop：录音结束  message：识别结果 |
| message | String | 语音识别结果 |

## 录音

### 开始录音录制

#### 方法名称

startRecording

#### 适用范围

页面浏览器

#### 参数说明：

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| sampleRateInHz | int | 44100 | 否 | 采样率: 44100 是目前的标 准，但是某些设备仍然支持 22050，16000，11025 (采样频率越高，能表现的频率 范围就越大，音质就会越 好，声音的还原度也更真实，但此同时带来的弊端是占有的内存资源也会越大) |
| channelConfig | int | 2 | 否 | 音频通道/声道: 1:单声道(左声道); 2 双声道 |
| requestPermission | bool | false | 否 | 是否申请系统权限 |
| permissionId | String | 无 | 否 | 唯一id（按id存储权限说明） |
| permissionDescription | String | 无 | 否 | 权限说明 |

**返回值data说明：**

{

    "code": 1,

    "message": "success",

}

示例

window.ado.startRecording().then((res) => {

console.log('startVoiceRecord res = ' + JSON.stringify(res))

// if (res.code === 0 && res.resultType === 'message') {

// this.voiceResult = res.result

// }

}).catch(error => {

console.log(error)

})

### 结束录音

#### 方法名称

stopRecording

#### 适用范围

页面浏览器

#### 参数说明：无

**返回值data说明：无**

返回值

{

    "code": 1,

    "message": "success",

}

示例

window.ado.stopRecording().then((res) => {

console.log('stopVoiceRecord')

}).catch(error => {

console.log(error)

})

### 监听语音识别结果

connectAdo (callback) {

if (window.ado) {

callback()

} else {

document.addEventListener('adoInit', function () {

callback()

}, false)

}

},

// 监听语音识别结果事件

mounted () {

this.connectAdo(() => {

window.ado.subscribe('voiceRecordEvent', (res) => {

console.log('handle voiceRecordEvent ', JSON.stringify(res))

if (res.type === 'pcm') {

this.voiceResult = res.message

}

})

})

},

// 取消监听

unMounted () {

this.connectAdo(() => {

window.ado.unSubscribe('voiceRecordEvent')

})

},

#### 事件返回值voiceRecordEvent说明

| **名称** | **类型** | **说明** |
| --- | --- | --- |
| type | String | start：录音开始  stop：录音结束  data：录音数据结果 |
| data | String | Base64格式的语音数据 |

## 悬浮图标

### 设置悬浮图标

#### 方法名称

showFloatMenu

#### 适用范围

页面浏览器

#### 参数说明：

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| imageUrl | String | 无 | 是 | 悬浮图标链接 |
| floatText | String | 无 | 否 | 悬浮轮播文字列表 |
| url | String | 无 | 是 | 点击图标需要打开的URL地址 1. 当h5AuthType为9时，填目标h5的url   1. 当h5AuthType为8时，填网格通应用链接 2. 当h5AuthType为0时，填小程序跳转路由 |
| h5AuthType | String | 9 | 是 | url类型:  9: app自带浏览器  8: 网格通应用 0: 小程序 |
| isKeyFunc | String | 0 | 是 | 1: 是关键业务，会校验是否需要刷脸刷证  0: 不是关键性业务 |

**返回值data说明：**

{

    "code": 1,

    "message": "success",

}

示例

window.ado.showFloatMenu({

“ImageUrl”: “xx”,

“floatTextList”:”XX”

}).then((res) => {

}).catch(error => {

console.log(error)

})

### 隐藏悬浮图标

#### 方法名称

hideFloatMenu

#### 适用范围

页面浏览器

#### 参数说明：

无

**返回值data说明：**

{

    "code": 1,

    "message": "success",

}

示例

window.ado.hideFloatMenu({

“ImageUrl”: “xx”,

“floatTextList”:”XX”

}).then((res) => {

console.log(res)

}).catch(error => {

console.log(error)

})

## 横竖屏切换

#### 方法名称

setOrientation

#### 适用范围

页面浏览器

#### 参数说明：

| **名称** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| orientation | String | 无 | 是 | landScape：横屏  portrait：竖屏 |

**返回值data说明：**

{

    "code": 1,

    "message": "success",

}

示例

window.ado.setOrientation({

“orientation”:”landScape”

}).then((res) => {

}).catch(error => {

console.log(error)

})

# 附录

## 小程序路由格式说明

小程序路由格式：

/mini/{appId}?path=**urlencode**(pages/index/index)&openType=refresh

appId：小程序id；

pages/index/index：小程序页面路由；  
openType: refresh会刷新小程序，否则只唤起已保活的小程序；