# 基础

**日志打印**

## [console](https://uniapp.dcloud.io/api/log?id=console)

向控制台打印日志信息。

### [debug](https://uniapp.dcloud.io/api/log?id=debug)

向控制台打印 debug 日志

注：App 端自定义组件模式下，debug 方法等同于 log 方法。

### [log](https://uniapp.dcloud.io/api/log?id=log)

向控制台打印 log 日志

### [info](https://uniapp.dcloud.io/api/log?id=info)

向控制台打印 info 日志

### [warn](https://uniapp.dcloud.io/api/log?id=warn)

向控制台打印 warn 日志

### [error](https://uniapp.dcloud.io/api/log?id=error)

向控制台打印 error 日志

注意：

* 不同平台对于 console 方法的支持存在差异，建议在开发过程中只使用文档中提到的方法。
* HBuilderX中有2个重要的代码块，敲clog：可直接输出console.log()；敲clogv：可输出console.log(": " + );，并且出现双光标，方便把变量名称和值同时打印出来。
* HBuilderX 1.9.7 以上的自定义组件模式，在App端支持打印对象信息到控制台。老版本可使用clogj代码块将json对象转为字符串打印出来。

## 定时器

setTimeout(callback, delay, rest)

设定一个定时器。在定时到期以后执行注册的回调函数

**参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| callback | Function | 是 | 回调函数 |
| delay | Number | 否 | 延迟的时间，函数的调用会在该延迟之后发生，单位 ms |
| rest | Any | 否 | param1, param2, ..., paramN 等附加参数，它们会作为参数传递给回调函数 |

**返回值**

| **返回值** | **类型** | **说明** |
| --- | --- | --- |
| timeoutID | Number | 定时器的编号，这个值可以传递给 [clearTimeout](https://uniapp.dcloud.io/api/timer?id=cleartimeout) 来取消该定时 |

clearTimeout(timeoutID)

取消由 setTimeout 设置的定时器。

**参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| timeoutID | Number | 是 | 要取消的定时器的 ID |

setInterval(callback, delay, rest)

设定一个定时器。按照指定的周期（以毫秒计）来执行注册的回调函数

**参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| callback | Function | 是 | 回调函数 |
| delay | Number | 否 | 延迟的时间，函数的调用会在该延迟之后发生，单位 ms |
| rest | Any | 否 | param1, param2, ..., paramN 等附加参数，它们会作为参数传递给回调函数 |

**返回值**

| **返回值** | **类型** | **说明** |
| --- | --- | --- |
| intervalID | Number | 定时器的编号，这个值可以传递给 [clearInterval](https://uniapp.dcloud.io/api/timer?id=clearinterval) 来取消该定时 |

clearInterval(intervalID)

取消由 setInterval 设置的定时器。

**参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| intervalID | Number | 是 | 要取消的定时器的 ID |

[uni.base64ToArrayBuffer(base64)](https://uniapp.dcloud.io/api/base64ToArrayBuffer?id=base64toarraybuffer)

将 Base64 字符串转成 ArrayBuffer 对象

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| √ | x | √ | x | x | x |

**参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| base64 | String | 是 | 要转化成 ArrayBuffer 对象的 Base64 字符串 |

**示例**

const base64 = 'test'

const arrayBuffer = uni.base64ToArrayBuffer(base64)

## [uni.arrayBufferToBase64(arrayBuffer)](https://uniapp.dcloud.io/api/arrayBufferToBase64?id=arraybuffertobase64)

将 ArrayBuffer 对象转成 Base64 字符串

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| √ | x | √ | x | x | x |

**参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| arrayBuffer | ArrayBuffer | 是 | 要转换成 Base64 字符串的 ArrayBuffer 对象 |

**示例**

const arrayBuffer = new Uint8Array([55, 55, 55])

const base64 = uni.arrayBufferToBase64(arrayBuffer)

## 生命周期

### [应用生命周期](https://uniapp.dcloud.io/api/lifecycle?id=%e5%ba%94%e7%94%a8%e7%94%9f%e5%91%bd%e5%91%a8%e6%9c%9f)

uni-app 支持如下应用生命周期函数：

| **函数名** | **说明** |
| --- | --- |
| onLaunch | 当uni-app 初始化完成时触发（全局只触发一次） |
| onShow | 当 uni-app 启动，或从后台进入前台显示 |
| onHide | 当 uni-app 从前台进入后台 |
| onUniNViewMessage | 对 nvue 页面发送的数据进行监听，可参考 [nvue 向 vue 通讯](https://uniapp.dcloud.io/use-weex?id=nvue-%e5%90%91-vue-%e9%80%9a%e8%ae%af) |

**注意**

* 应用生命周期仅可在App.vue中监听，在其它页面监听无效。
* onlaunch里进行页面跳转，如遇白屏报错，请参考<https://ask.dcloud.net.cn/article/35942>

### [页面生命周期](https://uniapp.dcloud.io/api/lifecycle?id=%e9%a1%b5%e9%9d%a2%e7%94%9f%e5%91%bd%e5%91%a8%e6%9c%9f)

uni-app 支持如下页面生命周期函数：

| **函数名** | **说明** | **平台差异说明** | **最低版本** |
| --- | --- | --- | --- |
| onLoad | 监听页面加载，其参数为上个页面传递的数据，参数类型为Object（用于页面传参），参考[示例](https://uniapp.dcloud.io/api/router?id=navigateto) |  |  |
| onShow | 监听页面显示。页面每次出现在屏幕上都触发，包括从下级页面点返回露出当前页面 |  |  |
| onReady | 监听页面初次渲染完成。注意如果渲染速度快，会在页面进入动画完成前触发 |  |  |
| onHide | 监听页面隐藏 |  |  |
| onUnload | 监听页面卸载 |  |  |
| onResize | 监听窗口尺寸变化 | App、微信小程序 |  |
| onPullDownRefresh | 监听用户下拉动作，一般用于下拉刷新，参考[示例](https://uniapp.dcloud.io/api/ui/pulldown) |  |  |
| onReachBottom | 页面滚动到底部的事件（不是scroll-view滚到底），常用于上拉加载下一页数据。如使用scroll-view导致页面级没有滚动，则触底事件不会被触发 |  |  |
| onTabItemTap | 点击 tab 时触发，参数为Object，具体见下方注意事项 | 微信小程序、百度小程序、H5、App（自定义组件模式） |  |
| onShareAppMessage | 用户点击右上角分享 | 微信小程序、百度小程序、头条小程序、支付宝小程序 |  |
| onPageScroll | 监听页面滚动，参数为Object |  |  |
| onNavigationBarButtonTap | 监听原生标题栏按钮点击事件，参数为Object | App、H5 |  |
| onBackPress | 监听页面返回，返回 event = {from:backbutton、 navigateBack} ，backbutton 表示来源是左上角返回按钮或 android 返回键；navigateBack表示来源是 uni.navigateBack ；详细说明及使用：[onBackPress 详解](https://ask.dcloud.net.cn/article/35120) | App、H5 |  |
| onNavigationBarSearchInputChanged | 监听原生标题栏搜索输入框输入内容变化事件 | App、H5 | 1.6.0 |
| onNavigationBarSearchInputConfirmed | 监听原生标题栏搜索输入框搜索事件，用户点击软键盘上的“搜索”按钮时触发。 | App、H5 | 1.6.0 |
| onNavigationBarSearchInputClicked | 监听原生标题栏搜索输入框点击事件 | App、H5 | 1.6.0 |

**注意**

* H5端onBackPress只能监听页面导航栏的返回按钮的点击事件，不能监听浏览器返回以及手机的返回

onPageScroll 参数说明：

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| scrollTop | Number | 页面在垂直方向已滚动的距离（单位px） |

onTabItemTap 参数说明：

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| index | String | 被点击tabItem的序号，从0开始 |
| pagePath | String | 被点击tabItem的页面路径 |
| text | String | 被点击tabItem的按钮文字 |

**注意**

* onTabItemTap常用于点击当前tabitem，滚动或刷新当前页面。如果是点击不同的tabitem，一定会触发页面切换。
* 如果想在App端实现点击某个tabitem不跳转页面，不能使用onTabItemTap，可以使用[plus.nativeObj.view](http://www.html5plus.org/doc/zh_cn/nativeobj.html)放一个区块盖住原先的tabitem，并拦截点击事件。
* onTabItemTap在App端，从HBuilderX 1.9 的自定义组件编译模式开始支持。

onNavigationBarButtonTap 参数说明：

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| index | Number | 原生标题栏按钮数组的下标 |

onBackPress 回调参数对象说明：

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| from | String | 触发返回行为的来源：'backbutton'——左上角导航栏按钮及安卓返回键；'navigateBack'——uni.navigateBack() 方法。 |

export default {

data() {

return {};

},

onBackPress(options) {

console.log('from:' + options.from)

}

}

**注意**

* nvue 页面的weex编译模式支持的生命周期，[参考](https://uniapp.dcloud.io/use-weex?id=%e7%94%9f%e5%91%bd%e5%91%a8%e6%9c%9f)。

# 网络

## 发起请求

### [uni.request(OBJECT)](https://uniapp.dcloud.io/api/request/request?id=request)

发起网络请求。

**在各个小程序平台运行时，网络相关的 API 在使用前需要配置域名白名单。**

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| url | String | 是 |  | 开发者服务器接口地址 |  |
| data | Object/String/ArrayBuffer | 否 |  | 请求的参数 | App（自定义组件编译模式）不支持ArrayBuffer类型 |
| header | Object | 否 |  | 设置请求的 header，header 中不能设置 Referer。 |  |
| method | String | 否 | GET | 有效值详见下方说明 |  |
| timeout | Number | 否 | 30000 | 超时时间，单位 ms | 支付宝小程序 |
| dataType | String | 否 | json | 如果设为 json，会尝试对返回的数据做一次 JSON.parse |  |
| responseType | String | 否 | text | 设置响应的数据类型。合法值：text、arraybuffer | App和支付宝小程序不支持 |
| sslVerify | Boolean | 否 | true | 验证 ssl 证书 | 仅App安卓端支持（HBuilderX 2.3.3+） |
| success | Function | 否 |  | 收到开发者服务成功返回的回调函数 |  |
| fail | Function | 否 |  | 接口调用失败的回调函数 |  |
| complete | Function | 否 |  | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**method 有效值**

必须大写，有效值在不同平台差异说明不同。

| **method** | **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| GET | √ | √ | √ | √ | √ | √ |
| POST | √ | √ | √ | √ | √ | √ |
| PUT | √ | √ | √ | x | √ | √ |
| DELETE | √ | √ | √ | x | √ | x |
| CONNECT | √ | √ | √ | x | x | x |
| HEAD | √ | √ | √ | x | √ | x |
| OPTIONS | √ | √ | √ | x | √ | x |
| TRACE | √ | √ | √ | x | x | x |

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| data | Object/String/ArrayBuffer | 开发者服务器返回的数据 |
| statusCode | Number | 开发者服务器返回的 HTTP 状态码 |
| header | Object | 开发者服务器返回的 HTTP Response Header |

**data 数据说明**

最终发送给服务器的数据是 String 类型，如果传入的 data 不是 String 类型，会被转换成 String。转换规则如下：

* 对于 GET 方法，会将数据转换为 query string。例如 { name: 'name', age: 18 } 转换后的结果是 name=name&age=18。
* 对于 POST 方法且 header['content-type'] 为 application/json 的数据，会进行 JSON 序列化。
* 对于 POST 方法且 header['content-type'] 为 application/x-www-form-urlencoded 的数据，会将数据转换为 query string。

**示例**

uni.request({

url: 'https://www.example.com/request', //仅为示例，并非真实接口地址。

data: {

text: 'uni.request'

},

header: {

'custom-header': 'hello' //自定义请求头信息

},

success: (res) => {

console.log(res.data);

this.text = 'request success';

}

});

**返回值**

如果希望返回一个 requestTask 对象，需要至少传入 success / fail / complete 参数中的一个。例如：

var requestTask = uni.request({

url: 'https://www.example.com/request', //仅为示例，并非真实接口地址。

complete: ()=> {}

});

requestTask.abort();

如果没有传入 success / fail / complete 参数，则会返回封装后的 Promise 对象：[Promise 封装](https://uniapp.dcloud.io/api/README?id=promise-%e5%b0%81%e8%a3%85)

通过 requestTask，可中断请求任务。

**requestTask 对象的方法列表**

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| abort |  | 中断请求任务 |
| offHeadersReceived |  | 取消监听 HTTP Response Header 事件，仅微信小程序平台支持，[文档详情](https://developers.weixin.qq.com/miniprogram/dev/api/RequestTask.offHeadersReceived.html) |
| onHeadersReceived |  | 监听 HTTP Response Header 事件。会比请求完成事件更早，仅微信小程序平台支持，[文档详情](https://developers.weixin.qq.com/miniprogram/dev/api/RequestTask.onHeadersReceived.html) |

**示例**

const requestTask = uni.request({

url: 'https://www.example.com/request', //仅为示例，并非真实接口地址。

data: {

name: 'name',

age: 18

},

success: function(res) {

console.log(res.data);

}

});

// 中断请求任务

requestTask.abort();

**Tips**

* 请求的 header 中 content-type 默认为 application/json。
* 网络请求的 超时时间 可以统一在 manifest.json 中配置 [networkTimeout](https://uniapp.dcloud.io/collocation/manifest?id=networktimeout)。
* H5 端本地调试需注意跨域问题，参考：[调试跨域问题解决方案](https://ask.dcloud.net.cn/article/35267)
* 注意由于百度小程序iOS客户端，请求失败时会进入fail回调，需要针对百度增加相应的处理以解决该问题。
* 注意非 H5 端不支持 cookie，服务器应避免验证 cookie。如果服务器无法修改，也可以使用一些模拟手段，比如这样的工具<https://github.com/charleslo1/weapp-cookie>
* 按照 W3C 规范，H5 端无法获取 response header 中 Set-Cookie、Set-Cookie2 这2个字段，对于跨域请求，允许获取的 response header 字段只限于“simple response header”和“Access-Control-Expose-Headers”（[详情](https://www.w3.org/TR/cors/#access-control-allow-credentials-response-header)）
* [uni-app 插件市场](https://ext.dcloud.net.cn/search?q=%E6%8B%A6%E6%88%AA%E5%99%A8)有flyio、axios等三方封装的拦截器可用
* 低版本手机自身不支持 ipv6，如果服务器仅允许 ipv6，会导致老手机无法正常运行或访问速度非常慢
* localhost、127.0.0.1等服务器地址，只能在电脑端运行，手机端连接时不能访问。请使用标准IP并保证手机能连接电脑网络
* debug 模式，安卓端暂时无法获取响应头，url中含有非法字符（如未编码为%20的空格）时会请求失败
* iOS App第一次安装启动后，会弹出是否允许联网的询问框，在用户点击同意前，调用联网API会失败。请注意判断这种情况。比如官方提供的新闻模板示例（HBuilderX新建项目可选择），会判断如果无法联网，则提供一个错误页，提示用户设置网络及下拉刷新重试。
* 良好体验的App，还会判断当前是否处于飞行模式（[参考](https://ext.dcloud.net.cn/plugin?id=594)）、是wifi还是3G（[参考](https://uniapp.dcloud.io/api/system/network)）
* 部分安卓设备，真机运行或debug模式下的网络，低于release模式很多。
* 使用一些比较小众的证书机构（如：CFCA OV OCA）签发的 ssl 证书在安卓设备请求会失败，因为这些机构的根证书不在系统内置根证书库，可以更换其他常见机构签发的证书（如：Let's Encrypt），或者配置 sslVerify 为 false 关闭 ssl 证书验证（不推荐）。
* 单次网络请求数据量建议控制在50K以下（仅指json数据，不含图片），过多数据应分页获取，以提升应用体验。

## 上下、下载

### [uni.uploadFile(OBJECT)](https://uniapp.dcloud.io/api/request/network-file?id=uploadfile)

将本地资源上传到开发者服务器，客户端发起一个 POST 请求，其中 content-type 为 multipart/form-data。  
如页面通过 [uni.chooseImage](https://uniapp.dcloud.io/api/media/image?id=chooseimage) 等接口获取到一个本地资源的临时文件路径后，可通过此接口将本地资源上传到指定服务器。另外选择和上传非图像、视频文件参考：<https://ask.dcloud.net.cn/article/35547>。

**在各个小程序平台运行时，网络相关的 API 在使用前需要配置域名白名单。**

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| url | String | 是 | 开发者服务器 url |  |
| files | Array | 否 | 需要上传的文件列表。**使用 files 时，filePath 和 name 不生效。** | App |
| fileType | String | 见平台差异说明 | 文件类型，image/video/audio | 仅支付宝小程序，且必填。 |
| filePath | String | 是 | 要上传文件资源的路径。 |  |
| name | String | 是 | 文件对应的 key , 开发者在服务器端通过这个 key 可以获取到文件二进制内容 |  |
| header | Object | 否 | HTTP 请求 Header, header 中不能设置 Referer。 |  |
| formData | Object | 否 | HTTP 请求中其他额外的 form data |  |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**注意**：

* App支持多文件上传，微信小程序只支持单文件上传，传多个文件需要反复调用本API。所以跨端的写法就是循环调用本API。
* hello uni-app中的客服反馈，支持多图上传。[uni-app插件市场](https://ext.dcloud.net.cn/)中也有多个封装的组件。
* App平台选择和上传非图像、视频文件，参考<https://ask.dcloud.net.cn/article/35547>
* 网络请求的 超时时间 可以统一在 manifest.json 中配置 [networkTimeout](https://uniapp.dcloud.io/collocation/manifest?id=networktimeout)。

**files参数说明**

files 参数是一个 file 对象的数组，file 对象的结构如下：

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| name | String | 否 | multipart 提交时，表单的项目名，默认为 file |
| uri | String | 是 | 文件的本地地址 |

Tip:

* 如果 name 不填或填的值相同，可能导致服务端读取文件时只能读取到一个文件。

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| data | String | 开发者服务器返回的数据 |
| statusCode | Number | 开发者服务器返回的 HTTP 状态码 |

**示例**

uni.chooseImage({

success: (chooseImageRes) => {

const tempFilePaths = chooseImageRes.tempFilePaths;

uni.uploadFile({

url: 'https://www.example.com/upload', //仅为示例，非真实的接口地址

filePath: tempFilePaths[0],

name: 'file',

formData: {

'user': 'test'

},

success: (uploadFileRes) => {

console.log(uploadFileRes.data);

}

});

}

});

**返回值**

如果希望返回一个 uploadTask 对象，需要至少传入 success / fail / complete 参数中的一个。例如：

var uploadTask = uni.uploadFile({

url: 'https://www.example.com/upload', //仅为示例，并非真实接口地址。

complete: ()=> {}

});

uploadTask.abort();

如果没有传入 success / fail / complete 参数，则会返回封装后的 Promise 对象：[Promise 封装](https://uniapp.dcloud.io/api/README?id=promise-%e5%b0%81%e8%a3%85)

通过 uploadTask，可监听上传进度变化事件，以及取消上传任务。

**uploadTask 对象的方法列表**

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| abort |  | 中断上传任务 |
| onProgressUpdate | callback | 监听上传进度变化 |
| onHeadersReceived | callback | 监听 HTTP Response Header 事件。会比请求完成事件更早,仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/UploadTask.onHeadersReceived.html) |
| offProgressUpdate | callback | 取消监听上传进度变化事件，仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/UploadTask.offProgressUpdate.html) |
| offHeadersReceived | callback | 取消监听 HTTP Response Header 事件，仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/UploadTask.offHeadersReceived.html) |

**onProgressUpdate 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| progress | Number | 上传进度百分比 |
| totalBytesSent | Number | 已经上传的数据长度，单位 Bytes |
| totalBytesExpectedToSend | Number | 预期需要上传的数据总长度，单位 Bytes |

**示例**

uni.chooseImage({

success: (chooseImageRes) => {

const tempFilePaths = chooseImageRes.tempFilePaths;

const uploadTask = uni.uploadFile({

url: 'https://www.example.com/upload', //仅为示例，非真实的接口地址

filePath: tempFilePaths[0],

name: 'file',

formData: {

'user': 'test'

},

success: (uploadFileRes) => {

console.log(uploadFileRes.data);

}

});

uploadTask.onProgressUpdate((res) => {

console.log('上传进度' + res.progress);

console.log('已经上传的数据长度' + res.totalBytesSent);

console.log('预期需要上传的数据总长度' + res.totalBytesExpectedToSend);

// 测试条件，取消上传任务。

if (res.progress > 50) {

uploadTask.abort();

}

});

}

});

### [uni.downloadFile(OBJECT)](https://uniapp.dcloud.io/api/request/network-file?id=downloadfile)

下载文件资源到本地，客户端直接发起一个 HTTP GET 请求，返回文件的本地临时路径。

**在各个小程序平台运行时，网络相关的 API 在使用前需要配置域名白名单。**

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| url | String | 是 | 下载资源的 url |
| header | Object | 否 | HTTP 请求 Header, header 中不能设置 Referer。 |
| success | Function | 否 | 下载成功后以 tempFilePath 的形式传给页面，res = {tempFilePath: '文件的临时路径'} |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**注：文件的临时路径，在应用本次启动期间可以正常使用，如需持久保存，需在主动调用**[uni.saveFile](https://uniapp.dcloud.io/api/file/file?id=savefile)**，才能在应用下次启动时访问得到。**

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| tempFilePath | String | 临时文件路径，下载后的文件会存储到一个临时文件 |
| statusCode | Number | 开发者服务器返回的 HTTP 状态码 |

**注意**

* 网络请求的 超时时间 可以统一在 manifest.json 中配置 [networkTimeout](https://uniapp.dcloud.io/collocation/manifest?id=networktimeout)。

**示例**

uni.downloadFile({

url: 'https://www.example.com/file/test', //仅为示例，并非真实的资源

success: (res) => {

if (res.statusCode === 200) {

console.log('下载成功');

}

}

});

**返回值**

如果希望返回一个 downloadTask 对象，需要至少传入 success / fail / complete 参数中的一个。例如：

var downloadTask = uni.downloadFile({

url: 'https://www.example.com/file/test', //仅为示例，并非真实接口地址。

complete: ()=> {}

});

downloadTask.abort();

如果没有传入 success / fail / complete 参数，则会返回封装后的 Promise 对象：[Promise 封装](https://uniapp.dcloud.io/api/README?id=promise-%e5%b0%81%e8%a3%85)

通过 downloadTask，可监听下载进度变化事件，以及取消下载任务。

**downloadTask 对象的方法列表**

| **方法** | **参数** | **说明** | **最低版本** |
| --- | --- | --- | --- |
| abort |  | 中断下载任务 | \* |
| onProgressUpdate | callback | 监听下载进度变化 | \* |
| onHeadersReceived | callback | 监听 HTTP Response Header 事件，会比请求完成事件更早,仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/DownloadTask.onHeadersReceived.html) |  |
| offProgressUpdate | callback | 取消监听下载进度变化事件，仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/DownloadTask.offProgressUpdate.html) |  |
| offHeadersReceived | callback | 取消监听 HTTP Response Header 事件，仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/DownloadTask.offHeadersReceived.html) |  |

**onProgressUpdate 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| progress | Number | 下载进度百分比 |
| totalBytesWritten | Number | 已经下载的数据长度，单位 Bytes |
| totalBytesExpectedToWrite | Number | 预期需要下载的数据总长度，单位 Bytes |

**示例**

const downloadTask = uni.downloadFile({

url: 'http://www.example.com/file/test', //仅为示例，并非真实的资源

success: (res) => {

if (res.statusCode === 200) {

console.log('下载成功');

}

}

});

downloadTask.onProgressUpdate((res) => {

console.log('下载进度' + res.progress);

console.log('已经下载的数据长度' + res.totalBytesWritten);

console.log('预期需要下载的数据总长度' + res.totalBytesExpectedToWrite);

// 测试条件，取消下载任务。

if (res.progress > 50) {

downloadTask.abort();

}

});

## WebSocket

### [uni.connectSocket(OBJECT)](https://uniapp.dcloud.io/api/request/websocket?id=connectsocket)

创建一个 [WebSocket](https://developer.mozilla.org/zh-CN/docs/Web/API/WebSocket) 连接。

**在各个小程序平台运行时，网络相关的 API 在使用前需要配置域名白名单。**

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| url | String | 是 | 服务器接口地址 | 小程序中必须是 wss:// 协议 |
| header | Object | 否 | HTTP Header , header 中不能设置 Referer | 小程序 |
| method | String | 否 | 默认是GET，有效值：OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT | 微信小程序 |
| protocols | Array<String> | 否 | 子协议数组 | H5、微信小程序、百度小程序、头条小程序 |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**示例代码**

uni.connectSocket({

url: 'wss://www.example.com/socket',

data() {

return {

x: '',

y: ''

};

},

header: {

'content-type': 'application/json'

},

protocols: ['protocol1'],

method: 'GET'

});

**返回值**

如果希望返回一个 [socketTask](https://uniapp.dcloud.io/api/request/socket-task) 对象，需要至少传入 success / fail / complete 参数中的一个。例如：

var socketTask = uni.connectSocket({

url: 'wss://www.example.com/socket', //仅为示例，并非真实接口地址。

complete: ()=> {}

});

如果没有传入 success / fail / complete 参数，则会返回封装后的 Promise 对象：[Promise 封装](https://uniapp.dcloud.io/api/README?id=promise-%e5%b0%81%e8%a3%85)

**注意事项**

* 网络请求的 超时时间 可以统一在 manifest.json 中配置 [networkTimeout](https://uniapp.dcloud.io/collocation/manifest?id=networktimeout)。
* App平台，2.2.6以下的版本，不支持 ArrayBuffer 类型的数据收发。老版本不愿升级也可以使用 [plus-websocket插件](https://ext.dcloud.net.cn/plugin?id=647) 插件替代。
* App平台自定义组件模式下，以及支付宝小程序下，所有 vue 页面只能使用一个 websocket 连接。App下可以使用 [plus-websocket](https://ext.dcloud.net.cn/plugin?id=647) 插件替代实现多链接。App平台，2.2.6+起支持多个socket链接，数量没有限制。
* 微信小程序平台1.7.0 及以上版本，最多可以同时存在5个WebSocket 连接。老版本只支持一个socket连接
* 百度小程序平台自基础库版本 1.9.4 及以后支持多个socket连接。老版本只支持一个socket连接
* QQ小程序平台最多支持同时存在5个socket链接

### [uni.onSocketOpen(CALLBACK)](https://uniapp.dcloud.io/api/request/websocket?id=onsocketopen)

监听WebSocket连接打开事件。

**CALLBACK 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| header | Object | 连接成功的 HTTP 响应 Header |

**示例代码：**

uni.connectSocket({

url: 'wss://www.example.com/socket'

});

uni.onSocketOpen(function (res) {

console.log('WebSocket连接已打开！');

});

### [uni.onSocketError(CALLBACK)](https://uniapp.dcloud.io/api/request/websocket?id=onsocketerror)

监听WebSocket错误。

**示例代码**

uni.connectSocket({

url: 'wss://www.example.com/socket'

});

uni.onSocketOpen(function (res) {

console.log('WebSocket连接已打开！');

});

uni.onSocketError(function (res) {

console.log('WebSocket连接打开失败，请检查！');

});

### [uni.sendSocketMessage(OBJECT)](https://uniapp.dcloud.io/api/request/websocket?id=sendsocketmessage)

通过 WebSocket 连接发送数据，需要先 [uni.connectSocket](https://uniapp.dcloud.io/api/request/websocket?id=connectsocket)，并在 [uni.onSocketOpen](https://uniapp.dcloud.io/api/request/websocket?id=onsocketopen) 回调之后才能发送。

**OBJECT 参数说明：**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| data | String/ArrayBuffer | 是 | 需要发送的内容 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例代码**

var socketOpen = false;

var socketMsgQueue = [];

uni.connectSocket({

url: 'wss://www.example.com/socket'

});

uni.onSocketOpen(function (res) {

socketOpen = true;

for (var i = 0; i < socketMsgQueue.length; i++) {

sendSocketMessage(socketMsgQueue[i]);

}

socketMsgQueue = [];

});

function sendSocketMessage(msg) {

if (socketOpen) {

uni.sendSocketMessage({

data: msg

});

} else {

socketMsgQueue.push(msg);

}

}

### [uni.onSocketMessage(CALLBACK)](https://uniapp.dcloud.io/api/request/websocket?id=onsocketmessage)

监听WebSocket接受到服务器的消息事件。

**CALLBACK 返回参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| data | String/ArrayBuffer | 服务器返回的消息 |

**示例代码：**

uni.connectSocket({

url: 'wss://www.example.com/socket'

});

uni.onSocketMessage(function (res) {

console.log('收到服务器内容：' + res.data);

});

### [uni.closeSocket(OBJECT)](https://uniapp.dcloud.io/api/request/websocket?id=closesocket)

关闭 WebSocket 连接。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| code | Number | 否 | 一个数字值表示关闭连接的状态号，表示连接被关闭的原因。如果这个参数没有被指定，默认的取值是1000 （表示正常连接关闭） |
| reason | String | 否 | 一个可读的字符串，表示连接被关闭的原因。这个字符串必须是不长于123字节的UTF-8 文本（不是字符） |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [uni.onSocketClose(CALLBACK)](https://uniapp.dcloud.io/api/request/websocket?id=onsocketclose)

监听WebSocket关闭。

uni.connectSocket({

url: 'wss://www.example.com/socket'

});

// 注意这里有时序问题，

// 如果 uni.connectSocket 还没回调 uni.onSocketOpen，而先调用 uni.closeSocket，那么就做不到关闭 WebSocket 的目的。

// 必须在 WebSocket 打开期间调用 uni.closeSocket 才能关闭。

uni.onSocketOpen(function () {

uni.closeSocket();

});

uni.onSocketClose(function (res) {

console.log('WebSocket 已关闭！');

});

## SocketTask

**SocketTask** 由 [uni.connectSocket()](https://uniapp.dcloud.io/api/request/websocket?id=connectsocket) 接口创建。

**平台差异说明**

支付宝小程序、头条小程序，没有明确的文档来具体说明这个对象，而是指向了 [Web Websocket](https://developer.mozilla.org/zh-CN/docs/Web/API/WebSocket) 对象。

### [SocketTask.onMessage(CALLBACK)](https://uniapp.dcloud.io/api/request/socket-task?id=sockettaskonmessage)

监听 WebSocket 接受到服务器的消息事件

**回调函数**

Function

WebSocket 接受到服务器的消息事件的回调函数

**回调函数中的参数**

Object

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| data | String/ArrayBuffer | 服务器返回的消息 |

### [SocketTask.send(OBJECT)](https://uniapp.dcloud.io/api/request/socket-task?id=sockettasksend)

通过 WebSocket 连接发送数据

**参数**

| **属性** | **类型** | **是否必填** | **说明** |
| --- | --- | --- | --- |
| data | String/ArrayBuffer | 是 | 需要发送的内容 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [SocketTask.close(OBJECT)](https://uniapp.dcloud.io/api/request/socket-task?id=sockettaskclose)

关闭 WebSocket 连接

**参数**

| **属性** | **类型** | **默认值** | **是否必填** | **说明** |
| --- | --- | --- | --- | --- |
| code | Number | 1000（表示正常关闭连接） | 否 | 一个数字值表示关闭连接的状态号，表示连接被关闭的原因。 |
| reason | String |  | 否 | 一个可读的字符串，表示连接被关闭的原因。 |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [SocketTask.onOpen(CALLBACK)](https://uniapp.dcloud.io/api/request/socket-task?id=sockettaskonopen)

监听 WebSocket 连接打开事件

**回调函数**

Function

WebSocket 连接打开事件的回调函数

**回调函数中的参数**

Object

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| data | String/ArrayBuffer | 服务器返回的消息 |

### [SocketTask.onClose(CALLBACK)](https://uniapp.dcloud.io/api/request/socket-task?id=sockettaskonclose)

监听 WebSocket 连接关闭事件

**回调函数**

Function

WebSocket 连接关闭事件的回调函数

### [SocketTask.onError(CALLBACK)](https://uniapp.dcloud.io/api/request/socket-task?id=sockettaskonerror)

监听 WebSocket 错误事件

**回调函数**

Function

WebSocket 错误事件的回调函数

**回调函数中的参数**

Object

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 错误信息 |

## mDNS服务

#### [mDNS 服务](https://uniapp.dcloud.io/api/request/mDNS?id=mdns-%e6%9c%8d%e5%8a%a1)

* 微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/wx.stopLocalServiceDiscovery.html)
* App端可在插件市场搜索相关插件[mDNS](https://ext.dcloud.net.cn/search?q=mdns)

## UDP通信

#### [UDP 通信](https://uniapp.dcloud.io/api/request/UDP?id=udp-%e9%80%9a%e4%bf%a1)

仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/network/udp/wx.createUDPSocket.html)

# 路由与页面跳转

#### [uni.navigateTo(OBJECT)](https://uniapp.dcloud.io/api/router?id=navigateto)

保留当前页面，跳转到应用内的某个页面，使用uni.navigateBack可以返回到原页面。

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| url | String | 是 |  | 需要跳转的应用内非 tabBar 的页面的路径 , 路径后可以带参数。参数与路径之间使用?分隔，参数键与参数值用=相连，不同参数用&分隔；如 'path?key=value&key2=value2'，path为下一个页面的路径，下一个页面的onLoad函数可得到传递的参数 | :- |
| animationType | String | 否 | pop-in | 窗口显示的动画效果，详见：[窗口动画](https://uniapp.dcloud.io/api/router?id=animation) | App |
| animationDuration | Number | 否 | 300 | 窗口动画持续时间，单位为 ms | App |
| success | Function | 否 |  | 接口调用成功的回调函数 |  |
| fail | Function | 否 |  | 接口调用失败的回调函数 |  |
| complete | Function | 否 |  | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**示例**

//在起始页面跳转到test.vue页面并传递参数

uni.navigateTo({

url: 'test?id=1&name=uniapp'

});

// 在test.vue页面接受参数

export default {

onLoad: function (option) { //option为object类型，会序列化上个页面传递的参数

console.log(option.id); //打印出上个页面传递的参数。

console.log(option.name); //打印出上个页面传递的参数。

}

}

url有长度限制，太长的字符串会传递失败，可使用[窗体通信](https://uniapp.dcloud.io/collocation/frame/communication)、[全局变量](https://ask.dcloud.net.cn/article/35021)，或encodeURIComponent等多种方式解决，如下为encodeURIComponent示例。

<navigator :url="'/pages/test/test?item='+ encodeURIComponent(JSON.stringify(item))"></navigator>

// 在test.vue页面接受参数

onLoad: function (option) {

const item = JSON.parse(decodeURIComponent(option.item));

}

**注意：**

* 页面跳转路径有层级限制，不能无限制跳转新页面
* 跳转到 tabBar 页面只能使用 switchTab 跳转
* 路由API的目标页面必须是在pages.json里注册的vue页面。如果想打开web url，在App平台可以使用 [plus.runtime.openURL](http://www.html5plus.org/doc/zh_cn/runtime.html" \l "plus.runtime.openURL" \t "_blank)或web-view组件；H5平台使用 window.open；小程序平台使用web-view组件（url需在小程序的联网白名单中）。在hello uni-app中有个组件ulink.vue已对多端进行封装，可参考。

#### [uni.redirectTo(OBJECT)](https://uniapp.dcloud.io/api/router?id=redirectto)

关闭当前页面，跳转到应用内的某个页面。

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| url | String | 是 | 需要跳转的应用内非 tabBar 的页面的路径，路径后可以带参数。参数与路径之间使用?分隔，参数键与参数值用=相连，不同参数用&分隔；如 'path?key=value&key2=value2' |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.redirectTo({

url: 'test?id=1'

});

**注意：**

* 跳转到 tabBar 页面只能使用 switchTab 跳转

#### [uni.reLaunch(OBJECT)](https://uniapp.dcloud.io/api/router?id=relaunch)

关闭所有页面，打开到应用内的某个页面。

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| url | String | 是 | 需要跳转的应用内页面路径 , 路径后可以带参数。参数与路径之间使用?分隔，参数键与参数值用=相连，不同参数用&分隔；如 'path?key=value&key2=value2'，如果跳转的页面路径是 tabBar 页面则不能带参数 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.reLaunch({

url: 'test?id=1'

});

export default {

onLoad: function (option) {

console.log(option.id);

}

}

#### [uni.switchTab(OBJECT)](https://uniapp.dcloud.io/api/router?id=switchtab)

跳转到 tabBar 页面，并关闭其他所有非 tabBar 页面。

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| url | String | 是 | 需要跳转的 tabBar 页面的路径（需在 pages.json 的 tabBar 字段定义的页面），路径后不能带参数 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

pages.json

{

"tabBar": {

"list": [{

"pagePath": "pages/index/index",

"text": "首页"

},{

"pagePath": "pages/other/other",

"text": "其他"

}]

}

}

other.vue

uni.switchTab({

url: '/pages/index/index'

});

#### [uni.navigateBack(OBJECT)](https://uniapp.dcloud.io/api/router?id=navigateback)

关闭当前页面，返回上一页面或多级页面。可通过 getCurrentPages() 获取当前的页面栈，决定需要返回几层。

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **默认值** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| delta | Number | 否 | 1 | 返回的页面数，如果 delta 大于现有页面数，则返回到首页。 |  |
| animationType | String | 否 | pop-out | 窗口关闭的动画效果，详见：[窗口动画](https://uniapp.dcloud.io/api/router?id=animation) | App |
| animationDuration | Number | 否 | 300 | 窗口关闭动画的持续时间，单位为 ms | App |

**示例**

// 注意：调用 navigateTo 跳转时，调用该方法的页面会被加入堆栈，而 redirectTo 方法则不会。见下方示例代码

// 此处是A页面

uni.navigateTo({

url: 'B?id=1'

});

// 此处是B页面

uni.navigateTo({

url: 'C?id=1'

});

// 在C页面内 navigateBack，将返回A页面

uni.navigateBack({

delta: 2

});

Tips：

* navigateTo, redirectTo 只能打开非 tabBar 页面。
* switchTab 只能打开 tabBar 页面。
* reLaunch 可以打开任意页面。
* 页面底部的 tabBar 由页面决定，即只要是定义为 tabBar 的页面，底部都有 tabBar。
* 不能在 App.vue 里面进行页面跳转。
* H5端页面刷新之后页面栈会消失，此时navigateBack不能返回，如果一定要返回可以使用history.back()。

**参考事项**

* 页面路由拦截和管理，插件市场有很多封装好的工具类，搜索[路由](https://ext.dcloud.net.cn/search?q=%E8%B7%AF%E7%94%B1)

#### [窗口动画](https://uniapp.dcloud.io/api/router?id=animation)

**本API仅App支持。小程序自身不支持自定义动画。H5的窗体动画可使用常规单页动画处理方案，见**[H5下单页动画示例](https://ext.dcloud.net.cn/plugin?id=659&tdsourcetag=s_pctim_aiomsg)

窗口的显示/关闭动画效果，支持在 API、组件、pages.json 中配置，优先级为：API = 组件 > pages.json。

##### [API](https://uniapp.dcloud.io/api/router?id=api)

有效的路由 API

* navigateTo
* navigateBack

uni.navigateTo({

url: '../test/test',

animationType: 'pop-in',

animationDuration: 200

});

uni.navigateBack({

delta: 1,

animationType: 'pop-out',

animationDuration: 200

});

##### [组件](https://uniapp.dcloud.io/api/router?id=%e7%bb%84%e4%bb%b6)

open-type 有效值

* navigateTo
* navigateBack

<navigator animation-type="pop-in" animation-duration="300" url="../test/test">navigator</navigator>

<navigator animation-type="pop-out" animation-duration="300" open-type="navigateBack" >navigator</navigator>

##### [pages.json](https://uniapp.dcloud.io/api/router?id=pagesjson)

pages.json 中配置的是窗口显示的动画

"style": {

"app-plus": {

"animationType": "fade-in",

"animationDuration": 300

}

}

显示动画与关闭动画，会有默认的对应规则。但是如果通过 API 或组件配置了窗口关闭的动画类型，则不会使用默认的类型。

| **显示动画** | **关闭动画** | **显示动画描述（关闭动画与之相反）** |
| --- | --- | --- |
| slide-in-right | slide-out-right | 新窗体从右侧进入 |
| slide-in-left | slide-out-left | 新窗体从左侧进入 |
| slide-in-top | slide-out-top | 新窗体从顶部进入 |
| slide-in-bottom | slide-out-bottom | 新窗体从底部进入 |
| pop-in | pop-out | 新窗体从左侧进入，且老窗体被挤压而出 |
| fade-in | fade-out | 新窗体从透明到不透明逐渐显示 |
| zoom-out | zoom-in | 新窗体从小到大缩放显示 |
| zoom-fade-out | zoom-fade-in | 新窗体从小到大逐渐放大并且从透明到不透明逐渐显示 |
| none | none | 无动画 |

详细的窗口动画说明，请参考：

* 窗口显示的动画：[AnimationTypeShow](http://www.html5plus.org/doc/zh_cn/webview.html#plus.webview.AnimationTypeShow)
* 窗口关闭的动画：[AnimationTypeClose](http://www.html5plus.org/doc/zh_cn/webview.html#plus.webview.AnimationTypeClose)

**注意**

* 纯nvue项目（render为native），窗体动画默认进入动画为popin，返回为pop-out。如果想修改动画类型，只能通过uni.navigateTo API修改，在组件或pages.json里配置动画类型无效
* 非纯nvue项目，App端窗体动画，默认进入动画为slider-in-right，默认返回动画为pop-out

# 数据缓存

#### [uni.setStorage(OBJECT)](https://uniapp.dcloud.io/api/storage/storage?id=setstorage)

将数据存储在本地缓存中指定的 key 中，会覆盖掉原来该 key 对应的内容，这是一个异步接口。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| key | String | 是 | 本地缓存中的指定的 key |
| data | Any | 是 | 需要存储的内容，只支持原生类型、及能够通过 JSON.stringify 序列化的对象 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.setStorage({

key: 'storage\_key',

data: 'hello',

success: function () {

console.log('success');

}

});

#### [uni.setStorageSync(KEY,DATA)](https://uniapp.dcloud.io/api/storage/storage?id=setstoragesync)

将 data 存储在本地缓存中指定的 key 中，会覆盖掉原来该 key 对应的内容，这是一个同步接口。

**参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| key | String | 是 | 本地缓存中的指定的 key |
| data | Any | 是 | 需要存储的内容，只支持原生类型、及能够通过 JSON.stringify 序列化的对象 |

try {

uni.setStorageSync('storage\_key', 'hello');

} catch (e) {

// error

}

#### [uni.getStorage(OBJECT)](https://uniapp.dcloud.io/api/storage/storage?id=getstorage)

从本地缓存中异步获取指定 key 对应的内容。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| key | String | 是 | 本地缓存中的指定的 key |
| success | Function | 是 | 接口调用的回调函数，res = {data: key对应的内容} |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| data | Any | key 对应的内容 |

**示例**

uni.getStorage({

key: 'storage\_key',

success: function (res) {

console.log(res.data);

}

});

#### [uni.getStorageSync(KEY)](https://uniapp.dcloud.io/api/storage/storage?id=getstoragesync)

从本地缓存中同步获取指定 key 对应的内容。

**参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| key | String | 是 | 本地缓存中的指定的 key |

**示例**

try {

const value = uni.getStorageSync('storage\_key');

if (value) {

console.log(value);

}

} catch (e) {

// error

}

#### [uni.getStorageInfo(OBJECT)](https://uniapp.dcloud.io/api/storage/storage?id=getstorageinfo)

异步获取当前 storage 的相关信息。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** |
| --- | --- | --- | --- | --- |
| HBuilderX 2.0.3+ | √ | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 是 | 接口调用的回调函数，详见返回参数说明 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| keys | Array＜String＞ | 当前 storage 中所有的 key |
| currentSize | Number | 当前占用的空间大小, 单位：kb |
| limitSize | Number | 限制的空间大小, 单位：kb |

**示例**

uni.getStorageInfo({

success: function (res) {

console.log(res.keys);

console.log(res.currentSize);

console.log(res.limitSize);

}

});

#### [uni.getStorageInfoSync()](https://uniapp.dcloud.io/api/storage/storage?id=getstorageinfosync)

同步获取当前 storage 的相关信息。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** |
| --- | --- | --- | --- | --- |
| HBuilderX 2.0.3+ | √ | √ | √ | √ |

**示例**

try {

const res = uni.getStorageInfoSync();

console.log(res.keys);

console.log(res.currentSize);

console.log(res.limitSize);

} catch (e) {

// error

}

#### [uni.removeStorage(OBJECT)](https://uniapp.dcloud.io/api/storage/storage?id=removestorage)

从本地缓存中异步移除指定 key。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| key | String | 是 | 本地缓存中的指定的 key |
| success | Function | 是 | 接口调用的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.removeStorage({

key: 'storage\_key',

success: function (res) {

console.log('success');

}

});

#### [uni.removeStorageSync(KEY)](https://uniapp.dcloud.io/api/storage/storage?id=removestoragesync)

从本地缓存中同步移除指定 key。

**参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| key | String | 是 | 本地缓存中的指定的 key |

**示例**

try {

uni.removeStorageSync('storage\_key');

} catch (e) {

// error

}

#### [uni.clearStorage()](https://uniapp.dcloud.io/api/storage/storage?id=clearstorage)

清理本地数据缓存。

**示例**

uni.clearStorage();

#### [uni.clearStorageSync()](https://uniapp.dcloud.io/api/storage/storage?id=clearstoragesync)

同步清理本地数据缓存。

**示例**

try {

uni.clearStorageSync();

} catch (e) {

// error

}

**注意**

uni-app的Storage在不同端的实现不同：

* H5端为localStorage，浏览器限制5M大小，是缓存概念，可能会被清理
* App端为原生的plus.storage，无大小限制，不是缓存，是持久化的
* 各个小程序端为其自带的storage api，数据存储生命周期跟小程序本身一致，即除用户主动删除或超过一定时间被自动清理，否则数据都一直可用。
* 微信小程序单个 key 允许存储的最大数据长度为 1MB，所有数据存储上限为 10MB。
* 支付宝小程序单条数据转换成字符串后，字符串长度最大200\*1024。同一个支付宝用户，同一个小程序缓存总上限为10MB。
* 百度、头条小程序文档未说明大小限制

除此之外，其他数据存储方案：

* H5端还支持websql、indexedDB、sessionStorage
* App端还支持[SQLite](https://www.html5plus.org/doc/zh_cn/sqlite.html" \t "_blank)、[IO文件](https://www.html5plus.org/doc/zh_cn/io.html" \t "_blank)等本地存储方案。

# 位置

## 获取位置

### [uni.getLocation(OBJECT)](https://uniapp.dcloud.io/api/location/location?id=getlocation)

获取当前的地理位置、速度。 在微信小程序中，当用户离开应用后，此接口无法调用，除非申请后台持续定位权限；当用户点击“显示在聊天顶部”时，此接口可继续调用。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| type | String | 否 | 默认为 wgs84 返回 gps 坐标，gcj02 返回国测局坐标，可用于 uni.openLocation 的坐标 |  |
| altitude | Boolean | 否 | 传入 true 会返回高度信息，由于获取高度需要较高精确度，会减慢接口返回速度 | App和头条小程序不支持 |
| geocode | Boolean | 否 | 默认false，是否解析地址信息 | 仅App平台支持 |
| success | Function | 是 | 接口调用成功的回调函数，返回内容详见返回参数说明。 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**success 返回参数说明**

| **参数** | **说明** |
| --- | --- |
| latitude | 纬度，浮点数，范围为-90~90，负数表示南纬 |
| longitude | 经度，浮点数，范围为-180~180，负数表示西经 |
| speed | 速度，浮点数，单位m/s |
| accuracy | 位置的精确度 |
| altitude | 高度，单位 m |
| verticalAccuracy | 垂直精度，单位 m（Android 无法获取，返回 0） |
| horizontalAccuracy | 水平精度，单位 m |
| [address](https://uniapp.dcloud.io/api/location/location?id=address) | 地址信息（仅App端支持，需配置geocode为true） |

**address 地址信息说明**

| **属性** | **类型** | **描述** | **说明** |
| --- | --- | --- | --- |
| country | String | 国家 | 如“中国”，如果无法获取此信息则返回undefined |
| province | String | 省份名称 | 如“北京市”，如果无法获取此信息则返回undefined |
| city | String | 城市名称 | 如“北京市”，如果无法获取此信息则返回undefined |
| district | String | 区（县）名称 | 如“朝阳区”，如果无法获取此信息则返回undefined |
| street | String | 街道信息 | 如“酒仙桥路”，如果无法获取此信息则返回undefined |
| streetNum | String | 获取街道门牌号信息 | 如“3号”，如果无法获取此信息则返回undefined |
| poiName | String | POI信息 | 如“电子城．国际电子总部”，如果无法获取此信息则返回undefined |
| postalCode | String | 邮政编码 | 如“100016”，如果无法获取此信息则返回undefined |
| cityCode | String | 城市代码 | 如“010”，如果无法获取此信息则返回undefined |

**示例**

uni.getLocation({

type: 'wgs84',

success: function (res) {

console.log('当前位置的经度：' + res.longitude);

console.log('当前位置的纬度：' + res.latitude);

}

});

#### [注意](https://uniapp.dcloud.io/api/location/location?id=%e6%b3%a8%e6%84%8f)

* H5：在较新的浏览器上，H5 端获取定位信息，要求部署在 **https** 服务上，本地预览（localhost）仍然可以使用 http 协议。
* H5：国产安卓手机上，H5若无法定位，检查手机是否开通位置服务、GPS，ROM是否给该浏览器位置权限、浏览器是否对网页弹出请求给予定位的询问框。
* H5：安卓手机在原生App内嵌H5时，无法定位需要原生App处理Webview。
* H5：移动端浏览器普遍仅支持GPS定位，在GPS信号弱的地方可能定位失败。
* H5：PC 设备使用 Chrome 浏览器的时候，位置信息是连接谷歌服务器获取的，国内用户可能获取位置信息失败。
* H5：使用坐标类型为 gcj02 时，需要配置腾讯地图 sdk 信息（manifest.json -> h5）。
* H5：微信公众号可使用微信js sdk，[详见](https://ask.dcloud.net.cn/article/35380" \t "_blank)
* App：Android由于谷歌服务被墙，想在国产手机上正常定位，需要向高德等三方服务商申请SDK资质，获取AppKey。云打包时需要在manifest的SDK配置中填写Appkey。在manifest可视化界面有详细申请指南，详见：<https://ask.dcloud.net.cn/article/29>。离线打包自行在原生工程中配置。注意包名、appkey、证书信息必须匹配。
* App：<map> 组件默认为国测局坐标gcj02，调用 uni.getLocation 返回结果传递给 <map> 组件时，需指定 type 为 gcj02。
* App：持续定位方案：iOS端可以申请持续定位权限，[参考](https://ask.dcloud.net.cn/article/12569)。Android如果进程被杀，代码无法执行。可以使用[unipush](https://ask.dcloud.net.cn/article/35622)，通过服务器激活App，执行透传消息，让App启动然后采集位置。Android上，即使自己写原生插件做后台进程，也很容易被杀，unipush是更合适的方案
* 小程序：api默认不返回详细地址中文描述。需要中文地址有2种方式：1、使用高德地图小程序sdk，在app和微信上都可以获得中文地址，[参考](http://ask.dcloud.net.cn/article/35070)。2、只考虑app，使用plus.geolocation也可以获取中文地址
* 可以通过用户授权API来判断用户是否给应用授予定位权限<https://uniapp.dcloud.io/api/other/authorize>

### [uni.chooseLocation(OBJECT)](https://uniapp.dcloud.io/api/location/location?id=chooselocation)

打开地图选择位置。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | x |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| keyword | String | 否 | 搜索关键字，仅App平台支持 |
| success | Function | 是 | 接口调用成功的回调函数，返回内容详见返回参数说明。 |
| fail | Function | 否 | 接口调用失败的回调函数（获取定位失败、用户取消等情况下触发） |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**注意**

* 因平台差异，如果SDK配置百度地图，需要设置keyword，才能显示相关地点
* nvue下只支持高德地图，不支持百度地图
* HBuilderX 2.4.0+ 非 weex 编译模式仅支持高德地图

**success 返回参数说明**

| **参数** | **说明** |
| --- | --- |
| name | 位置名称 |
| address | 详细地址 |
| latitude | 纬度，浮点数，范围为-90~90，负数表示南纬 |
| longitude | 经度，浮点数，范围为-180~180，负数表示西经 |

**示例**

uni.chooseLocation({

success: function (res) {

console.log('位置名称：' + res.name);

console.log('详细地址：' + res.address);

console.log('纬度：' + res.latitude);

console.log('经度：' + res.longitude);

}

});

**注意**

* 不同端，使用地图选择时基于的底层地图引擎不一样，如微信小程序和H5是腾讯地图，App和阿里小程序是高德地图（注意app端不支持百度地图），百度小程序是百度地图，详见地图map组件的使用注意事项
* 微信内置浏览器中可使用微信js sdk，[详见](https://ask.dcloud.net.cn/article/35380" \t "_blank)
* chooseLocation属于封装型API，开发者若觉得不够灵活，可自行基于原始的map组件进行封装
* 若Android App端位置不准，见上文uni.getLocation的注意事项

## 查看位置

### [uni.openLocation(OBJECT)](https://uniapp.dcloud.io/api/location/open-location?id=openlocation)

使用应用内置地图查看位置。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | √ | x |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| latitude | Float | 是 | 纬度，范围为-90~90，负数表示南纬，使用 gcj02 国测局坐标系 |  |
| longitude | Float | 是 | 经度，范围为-180~180，负数表示西经，使用 gcj02 国测局坐标系 |  |
| scale | Int | 否 | 缩放比例，范围5~18，默认为18 | 微信小程序 |
| name | String | 否 | 位置名 |  |
| address | String | 否 | 地址的详细说明 |  |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**示例**

uni.getLocation({

type: 'gcj02', //返回可以用于uni.openLocation的经纬度

success: function (res) {

const latitude = res.latitude;

const longitude = res.longitude;

uni.openLocation({

latitude: latitude,

longitude: longitude,

success: function () {

console.log('success');

}

});

}

});

**Tips**

* 本API是一个非原子的封装界面，开发者如有定制需求，可自己做页面实现类似功能。
* 若定位不准，参考[uni.getLocation文档的注意事项](https://uniapp.dcloud.io/api/location/location)
* <map> 组件默认为国测局坐标，调用 uni.getLocation 返回结果传递给 <map> 组件时，需指定 type 为 gcj02。
* App 端使用地图组件需要向高德等三方服务商申请SDK资质，获取AppKey，打包时需要在manifest的SDK配置中填写Appkey。在manifest可视化界面有详细申请指南，详见：<https://ask.dcloud.net.cn/article/29>。
* iOS端 如需使用第三方地图进行导航，需要在 manifest.json 文件内增加 urlschemewhitelist 节点，在线打包即可
* 不同平台打开的地图不一样，App端是高德地图（不支持百度地图）、H5端是腾讯地图、微信小程序端是腾讯地图、百度小程序是百度地图、阿里小程序是高德地图、头条小程序是高德地图。

{

"app-plus": {

"distribute": {

"ios": {

"urlschemewhitelist": [

"baidumap",

"iosamap",

"qqmap"

]

}

}

}

}

* 点击返回也会进入 fail 回调中

## 地图组件控制

### [uni.createMapContext(mapId,this)](https://uniapp.dcloud.io/api/location/map?id=createmapcontext)

创建并返回 map 上下文 mapContext 对象。在自定义组件下，第二个参数传入组件实例this，以操作组件内 <map> 组件。

**注意：uni.createMapContext(mapId, this)**

* app-nvue 平台 2.2.5+ 支持 uni.createMapContext(mapId, this)
* app-nvue 平台 2.2.5- 需要同时设置组件属性id和ref <map id="map1" ref="map1"></map>，或者直接使用 ref，例如 this.$refs.map1

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | x |

mapContext

mapContext 通过 mapId 跟一个 <map> 组件绑定，通过它可以操作对应的 <map> 组件。

**mapContext 对象的方法列表**

| **方法** | **参数** | **说明** | **平台差异说明** | **最低版本** |
| --- | --- | --- | --- | --- |
| getCenterLocation | OBJECT | 获取当前地图中心的经纬度，返回的是 gcj02 坐标系，可以用于 [uni.openLocation](https://uniapp.dcloud.io/api/location/location?id=getlocation) |  |  |
| moveToLocation |  | 将地图中心移动到当前定位点，需要配合map组件的show-location使用 |  |  |
| translateMarker | OBJECT | 平移marker，带动画 | app-nvue 2.1.5+、微信小程序带动画 |  |
| includePoints | OBJECT | 缩放视野展示所有经纬度 | app-nvue 2.1.5+ |  |
| getRegion | OBJECT | 获取当前地图的视野范围 |  |  |
| getScale | OBJECT | 获取当前地图的缩放级别 |  |  |
| $getAppMap |  | 获取原生地图对象 [plus.maps.Map](https://www.html5plus.org/doc/zh_cn/maps.html" \l "plus.maps.Map" \t "_blank) | app-vue自定义组件模式 | 1.9.3 |

$getAppMap() 注意事项：

* 在页面中，必须在 onReady 中调用。
* 在组件中，必须在 mounted 中调用。
* nvue没有$getAppMap()，请使用createMapContext
* uni-app中使用原生地图无需提供占位div，得到$getAppMap()后直接js使用即可。

**getCenterLocation 的 OBJECT 参数列表**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调函数 ，res = { longitude: "经度", latitude: "纬度"} |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**translateMarker 的 OBJECT 参数列表**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| markerId | Number | 是 | 指定 marker |
| destination | Object | 是 | 指定 marker 移动到的目标点 |
| autoRotate | Boolean | 是 | 移动过程中是否自动旋转 marker |
| rotate | Number | 是 | marker 的旋转角度 |
| duration | Number | 否 | 动画持续时长，默认值1000ms，平移与旋转分别计算 |
| animationEnd | Function | 否 | 动画结束回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |

**includePoints 的 OBJECT 参数列表**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| points | Array | 是 | 要显示在可视区域内的坐标点列表，[{latitude, longitude}] |
| padding | Array | 否 | 坐标点形成的矩形边缘到地图边缘的距离，单位像素。格式为[上，右，下，左]，安卓上只能识别数组第一项，上下左右的padding一致。开发者工具暂不支持padding参数。 |

**getRegion 的 OBJECT 参数列表**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调函数，res = {southwest, northeast}，西南角与东北角的经纬度 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**getScale 的 OBJECT 参数列表**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调函数，res = {scale} |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

## [mapSearch 模块(module)](https://uniapp.dcloud.io/api/location/map?id=mapsearch-%e6%a8%a1%e5%9d%97)

#### [reverseGeocode(Object,callback)](https://uniapp.dcloud.io/api/location/map?id=reversegeocode)

**反向地理编码**

##### [Object](https://uniapp.dcloud.io/api/location/map?id=object)

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| point | Object |  | 是 | {latitude: 纬度, longitude: 经度} |

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/location/map?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e)

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| type | String | "success" 表示成功， "fail" 表示失败 |
| code | Number | 成功返回 0,失败返回相应 code 码 |
| message | String | 失败描述 |
| address | String | 查询后地址 （成功时返回） |

#### [poiSearchNearBy（Object,callback);](https://uniapp.dcloud.io/api/location/map?id=poisearchnearby%ef%bc%88objectcallback)

**周边检索**

##### [Object](https://uniapp.dcloud.io/api/location/map?id=object-1)

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| point | Object |  | 是 | 检索的中心点坐标 {latitude: 纬度, longitude: 经度} |
| key | String |  | 是 | 搜索关键字 |
| radius | Number | 3000 | 否 | 检索的半径，单位为米 |
| index | Number | 1 | 否 | 要获取检索结果的页号索引 |

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/location/map?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e-1)

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| type | String | "success" 表示成功， "fail" 表示失败 |
| code | Number | 成功返回 0,失败返回相应 code 码 |
| message | String | 失败描述 |
| totalNumber | Number | 返回的POI数目 |
| currentNumber | Number | 当前页POI数目 |
| pageNumber | Number | 页数 |
| pageIndex | Number | 当前页号索引 |
| poiList | Array.<poiObject> | POI信息数组 |

##### [poiObject](https://uniapp.dcloud.io/api/location/map?id=poiobject)

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| location | Object | {latitude: 纬度, longitude: 经度} |
| name | String | 名称 |
| type | String | 类型 |
| distance | Number | 距离（单位米） |
| address | String | 地址 |

**Tips**

* App端使用map，nvue比vue更强大。
* App端vue页面默认为高德地图，也可以选择百度地图。但app-nvue只有高德地图，没有百度地图。以及地图选择api，只支持高德地图。
* H5 端获取定位信息，需要部署在 **https** 服务上，本地预览（localhost）仍然可以使用 http 协议。
* 无 GPS 模块的 PC 设备使用 Chrome 浏览器的时候，位置信息是连接谷歌服务器获取的，国内用户可能获取位置信息失败。
* App 端使用地图组件需要向高德或百度等三方服务商申请SDK资质，获取AppKey，打包时需要在manifest的SDK配置中填写Appkey。在manifest可视化界面有详细申请指南，详见：<https://ask.dcloud.net.cn/article/29>
* <map> 组件默认为国测局坐标，调用 uni.getLocation 返回结果传递给 <map> 组件时，需指定 type 为 gcj02。

# 媒体

## 图片

### [uni.chooseImage(OBJECT)](https://uniapp.dcloud.io/api/media/image?id=chooseimage)

从本地相册选择图片或使用相机拍照。

App端如需要更丰富的相机拍照API（如直接调用前置摄像头），参考[plus.camera](https://www.html5plus.org/doc/zh_cn/camera.html)

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| count | Number | 否 | 最多可以选择的图片张数，默认9 | 见下方说明 |
| sizeType | Array<String> | 否 | original 原图，compressed 压缩图，默认二者都有 | App、微信小程序、支付宝小程序、百度小程序 |
| sourceType | Array<String> | 否 | album 从相册选图，camera 使用相机，默认二者都有。如需直接开相机或直接选相册，请只使用一个选项 |  |
| success | Function | 是 | 成功则返回图片的本地文件路径列表 tempFilePaths |  |
| fail | Function | 否 | 接口调用失败的回调函数 | 小程序、App |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**Tips**

* count 值在 H5 平台的表现，基于浏览器本身的规范。目前测试的结果来看，只能限制单选/多选，并不能限制数量。并且，在实际的手机浏览器很少有能够支持多选的。
* sourceType 在H5端对应input的capture属性，设置为['album']无效，依然可以使用相机。
* 可以通过用户授权API来判断用户是否给应用授予相册或摄像头的访问权限<https://uniapp.dcloud.io/api/other/authorize>
* App端如需选择非媒体文件，可在插件市场搜索[文件选择](https://ext.dcloud.net.cn/search?q=%E6%96%87%E4%BB%B6%E9%80%89%E6%8B%A9)，其中Android端可以使用Native.js，无需原生插件，而iOS端需要原生插件。

**注：文件的临时路径，在应用本次启动期间可以正常使用，如需持久保存，需在主动调用**[uni.saveFile](https://uniapp.dcloud.io/api/file/file?id=savefile)**，在应用下次启动时才能访问得到。**

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| tempFilePaths | Array<String> | 图片的本地文件路径列表 |
| tempFiles | Array<Object> | 图片的本地文件列表，每一项是一个 File 对象 |

**File 对象结构如下**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| path | String | 本地文件路径 |
| size | Number | 本地文件大小，单位：B |
| name | String | 包含扩展名的文件名称，仅H5支持 |

**示例**

uni.chooseImage({

count: 6, //默认9

sizeType: ['original', 'compressed'], //可以指定是原图还是压缩图，默认二者都有

sourceType: ['album'], //从相册选择

success: function (res) {

console.log(JSON.stringify(res.tempFilePaths));

}

});

### [uni.previewImage(OBJECT)](https://uniapp.dcloud.io/api/media/image?id=unipreviewimageobject)

预览图片。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| current | String/Number | 详见下方说明 | 详见下方说明 |  |
| urls | Array<String> | 是 | 需要预览的图片链接列表 |  |
| indicator | String | 否 | 图片指示器样式，可取值："default" - 底部圆点指示器； "number" - 顶部数字指示器； "none" - 不显示指示器。 | App |
| loop | Boolean | 否 | 是否可循环预览，默认值为 false | App |
| longPressActions | Object | 否 | 长按图片显示操作菜单，如不填默认为**保存相册** | App 1.9.5+ |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**current 参数说明**

**1.9.5+ 支持传图片在 urls 中的索引值**

current 为当前显示图片的链接/索引值，不填或填写的值无效则为 urls 的第一张。**App平台在 1.9.5至1.9.8之间，current为必填。不填会报错**

注意，当 urls 中有重复的图片链接时：

* 传链接，预览结果始终显示该链接在 urls 中第一次出现的位置。
* 传索引值，在微信/百度/头条小程序平台，会过滤掉传入的 urls 中该索引值之前与其对应图片链接重复的值。其它平台会保留原始的 urls 不会做去重处理。

举例说明：

一组图片 [A, B1, C, B2, D]，其中 B1 与 B2 的图片链接是一样的。

* 传 B2 的链接，预览的结果是 B1，前一张是 A，下一张是 C。
* 传 B2 的索引值 3，预览的结果是 B2，前一张是 C，下一张是 D。此时在微信/百度/头条小程序平台，最终传入的 urls 是 [A, C, B2, D]，过滤掉了与 B2 重复的 B1。

**longPressActions 参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| itemList | Array<String> | 是 | 按钮的文字数组 |
| itemColor | String | 否 | 按钮的文字颜色，字符串格式，默认为"#000000" |
| success | Function | 否 | 接口调用成功的回调函数，详见返回参数说明 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| index | Number | 用户长按图片的索引值 |
| tapIndex | Number | 用户点击按钮列表的索引值 |

**示例**

// 从相册选择6张图

uni.chooseImage({

count: 6,

sizeType: ['original', 'compressed'],

sourceType: ['album'],

success: function(res) {

// 预览图片

uni.previewImage({

urls: res.tempFilePaths,

longPressActions: {

itemList: ['发送给朋友', '保存图片', '收藏'],

success: function(data) {

console.log('选中了第' + (data.tapIndex + 1) + '个按钮,第' + (data.index + 1) + '张图片');

},

fail: function(err) {

console.log(err.errMsg);

}

}

});

}

});

### [uni.getImageInfo(OBJECT)](https://uniapp.dcloud.io/api/media/image?id=getimageinfo)

获取图片信息。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| src | String | 是 | 图片的路径，可以是相对路径，临时文件路径，存储文件路径，网络图片路径 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数名** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| width | Number | 图片宽度，单位px |  |
| height | Number | 图片高度，单位px |  |
| path | String | 返回图片的本地路径 |  |
| orientation | String | 返回图片的方向，有效值见下表 | App、小程序 |
| type | String | 返回图片的格式 | App、小程序 |

**orientation 参数说明**

| **枚举值** | **说明** |
| --- | --- |
| up | 默认 |
| down | 180度旋转 |
| left | 逆时针旋转90度 |
| right | 顺时针旋转90度 |
| up-mirrored | 同up，但水平翻转 |
| down-mirrored | 同down，但水平翻转 |
| left-mirrored | 同left，但垂直翻转 |
| right-mirrored | 同right，但垂直翻转 |

**示例**

uni.chooseImage({

count: 1,

sourceType: ['album'],

success: function (res) {

uni.getImageInfo({

src: res.tempFilePaths[0],

success: function (image) {

console.log(image.width);

console.log(image.height);

}

});

}

});

### [uni.saveImageToPhotosAlbum(OBJECT)](https://uniapp.dcloud.io/api/media/image?id=saveimagetophotosalbum)

保存图片到系统相册。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| filePath | String | 是 | 图片文件路径，可以是临时文件路径也可以是永久文件路径，不支持网络图片路径 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 调用结果 |

**注意**

* 可以通过用户授权API来判断用户是否给应用授予相册的访问权限<https://uniapp.dcloud.io/api/other/authorize>

**示例代码：**

uni.chooseImage({

count: 1,

sourceType: ['camera'],

success: function (res) {

uni.saveImageToPhotosAlbum({

filePath: res.tempFilePaths[0],

success: function () {

console.log('save success');

}

});

}

});

### Uni.compresslmage(OBHECT)

压缩图片接口，可选压缩质量

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| 1.9.7+ [自定义组件编译模式](https://ask.dcloud.net.cn/article/35843) | x | √ | √ | √(基础库版本>=3.110.3) | √ |

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| src | String |  | 是 | 图片路径，图片的路径，可以是相对路径、临时文件路径、存储文件路径 |
| quality | Number | 80 | 否 | 压缩质量，范围0～100，数值越小，质量越低，压缩率越高（仅对jpg有效） |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| tempFilePath | String | 压缩后图片的临时文件路径 |

**示例代码：**

uni.compressImage({

src: '/static/logo.jpg',

quality: 80,

success: res => {

console.log(res.tempFilePath)

}

})

### Wx.chooseMessageFile(OBJECT)

从微信聊天会话中选择文件。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| x | x | √ | x | x | x |

## 录音管理

### [uni.getRecorderManager()](https://uniapp.dcloud.io/api/media/record-manager?id=getrecordermanager)

获取**全局唯一**的录音管理器 recorderManager。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | x | √ | √ | √ |

**recorderManager 对象的方法列表**

| **方法** | **参数** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| start | options | 开始录音 |  |
| pause |  | 暂停录音 |  |
| resume |  | 继续录音 |  |
| stop |  | 停止录音 |  |
| onStart | callback | 录音开始事件 |  |
| onPause | callback | 录音暂停事件 |  |
| onStop | callback | 录音停止事件，会回调文件地址 |  |
| onFrameRecorded | callback | 已录制完指定帧大小的文件，会回调录音分片结果数据。如果设置了 frameSize ，则会回调此事件 | App 暂不支持 |
| onError | callback | 录音错误事件, 会回调错误信息 |  |

**start(options) 说明**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| duration | Number | 否 | 指定录音的时长，单位 ms ，如果传入了合法的 duration ，在到达指定的 duration 后会自动停止录音，最大值 600000（10 分钟）,默认值 60000（1 分钟） |
| sampleRate | Number | 否 | 采样率，有效值 8000/16000/44100 |
| numberOfChannels | Number | 否 | 录音通道数，有效值 1/2 |
| encodeBitRate | Number | 否 | 编码码率，有效值见下表格 |
| format | String | 否 | 音频格式，有效值 aac/mp3 |
| frameSize | String | 否 | 指定帧大小，单位 KB。传入 frameSize 后，每录制指定帧大小的内容后，会回调录制的文件内容，不指定则不会回调。暂仅支持 mp3 格式。 |

其中，采样率和码率有一定要求，具体有效值如下：

| **采样率** | **编码码率** |
| --- | --- |
| 8000 | 16000 ~ 48000 |
| 11025 | 16000 ~ 48000 |
| 12000 | 24000 ~ 64000 |
| 16000 | 24000 ~ 96000 |
| 22050 | 32000 ~ 128000 |
| 24000 | 32000 ~ 128000 |
| 32000 | 48000 ~ 192000 |
| 44100 | 64000 ~ 320000 |
| 48000 | 64000 ~ 320000 |

**onStop(callback) 回调结果说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| tempFilePath | String | 录音文件的临时路径 |

**onFrameRecorded(callback) 回调结果说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| frameBuffer | ArrayBuffer | 录音分片结果数据 |
| isLastFrame | Boolean | 当前帧是否正常录音结束前的最后一帧 |

**onError(callback) 回调结果说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 错误信息 |

**注意**

* 可以通过用户授权API来判断用户是否给应用授予麦克风的访问权限<https://uniapp.dcloud.io/api/other/authorize>

**示例**

<template>

<view>

<button @tap="startRecord">开始录音</button>

<button @tap="endRecord">停止录音</button>

<button @tap="playVoice">播放录音</button>

</view>

</template>

const recorderManager = uni.getRecorderManager();

const innerAudioContext = uni.createInnerAudioContext();

innerAudioContext.autoplay = true;

export default {

data: {

text: 'uni-app',

voicePath: ''

},

onLoad() {

let self = this;

recorderManager.onStop(function (res) {

console.log('recorder stop' + JSON.stringify(res));

self.voicePath = res.tempFilePath;

});

},

methods: {

startRecord() {

console.log('开始录音');

recorderManager.start();

},

endRecord() {

console.log('录音结束');

recorderManager.stop();

},

playVoice() {

console.log('播放录音');

if (this.voicePath) {

innerAudioContext.src = this.voicePath;

innerAudioContext.play();

}

}

}

}

## 背景音频播放管理

### [uni.getBackgroundAudioManager()](https://uniapp.dcloud.io/api/media/background-audio-manager?id=getbackgroundaudiomanager)

获取**全局唯一**的背景音频管理器 backgroundAudioManager。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | x | √ | √ | √ |

**backgroundAudioManager 对象的属性列表**

| **属性** | **类型** | **说明** | **只读** |
| --- | --- | --- | --- |
| duration | Number | 当前音频的长度（单位：s），只有在当前有合法的 src 时返回 | 是 |
| currentTime | Number | 当前音频的播放位置（单位：s），只有在当前有合法的 src 时返回 | 是 |
| paused | Boolean | 当前是是否暂停或停止状态，true 表示暂停或停止，false 表示正在播放 | 是 |
| src | String | 音频的数据源，默认为空字符串，**当设置了新的 src 时，会自动开始播放，**目前支持的格式有 m4a, aac, mp3, wav | 否 |
| startTime | Number | 音频开始播放的位置（单位：s） | 否 |
| buffered | Number | 音频缓冲的时间点，仅保证当前播放时间点到此时间点内容已缓冲。 | 是 |
| title | String | 音频标题，用于做原生音频播放器音频标题。原生音频播放器中的分享功能，分享出去的卡片标题，也将使用该值。 | 否 |
| epname | String | 专辑名，原生音频播放器中的分享功能，分享出去的卡片简介，也将使用该值。 | 否 |
| singer | String | 歌手名，原生音频播放器中的分享功能，分享出去的卡片简介，也将使用该值。 | 否 |
| coverImgUrl | String | 封面图url，用于做原生音频播放器背景图。原生音频播放器中的分享功能，分享出去的卡片配图及背景也将使用该图。 | 否 |
| webUrl | String | 页面链接，原生音频播放器中的分享功能，分享出去的卡片简介，也将使用该值。 | 否 |
| protocol | String | 音频协议。默认值为 'http'，设置 'hls' 可以支持播放 HLS 协议的直播音频 | 否 |

**backgroundAudioManager 对象的方法列表**

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| play |  | 播放 |
| pause |  | 暂停 |
| stop |  | 停止 |
| seek | position | 跳转到指定位置，单位 s |
| onCanplay | callback | 背景音频进入可以播放状态，但不保证后面可以流畅播放 |
| onPlay | callback | 背景音频播放事件 |
| onPause | callback | 背景音频暂停事件 |
| onStop | callback | 背景音频停止事件 |
| onEnded | callback | 背景音频自然播放结束事件 |
| onTimeUpdate | callback | 背景音频播放进度更新事件 |
| onPrev | callback | 用户在系统音乐播放面板点击上一曲事件（iOS only） |
| onNext | callback | 用户在系统音乐播放面板点击下一曲事件（iOS only） |
| onError | callback | 背景音频播放错误事件 |
| onWaiting | callback | 音频加载中事件，当音频因为数据不足，需要停下来加载时会触发 |

errCode 说明

| **errCode** | **说明** |
| --- | --- |
| 10001 | 系统错误 |
| 10002 | 网络错误 |
| 10003 | 文件错误 |
| 10004 | 格式错误 |
| -1 | 未知错误 |

**示例**

const bgAudioMannager = uni.getBackgroundAudioManager();

bgAudioMannager.title = '致爱丽丝';

bgAudioMannager.singer = '暂无';

bgAudioMannager.coverImgUrl = 'https://img-cdn-qiniu.dcloud.net.cn/uniapp/audio/music.jpg';

bgAudioMannager.src = 'https://img-cdn-qiniu.dcloud.net.cn/uniapp/audio/music.mp3';

**注意** 因为背景音频播放耗费手机电量，所以平台都有管控，需在manifest中填写申请。

* ios App平台，背景播放需在manifest.json -> app-plus -> distribute -> ios 节点添加 "UIBackgroundModes":["audio"] 才能保证音乐可以后台播放（打包成ipa生效）
* 小程序平台，需在manifest.json 对应的小程序节点下，填写"requiredBackgroundModes": ["audio"]。发布小程序时平台会审核
* Android App端默认不会在通知栏出现音量控制，如需此功能，需要在插件市场单独下载原生插件，[详见](https://ext.dcloud.net.cn/search?q=%E9%80%9A%E7%9F%A5%E6%A0%8F+%E9%9F%B3%E4%B9%90%E6%8E%A7%E5%88%B6)

## 音频组件控制

### [uni.createInnerAudioContext()](https://uniapp.dcloud.io/api/media/audio-context?id=createinneraudiocontext)

创建并返回内部 audio 上下文 innerAudioContext 对象。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | x | √ | √ | √ |

**innerAudioContext 对象的属性列表**

| **属性** | **类型** | **说明** | **只读** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| src | String | 音频的数据链接，用于直接播放。 | 否 |  |
| startTime | Number | 开始播放的位置（单位：s），默认 0 | 否 |  |
| autoplay | Boolean | 是否自动开始播放，默认 false | 否 | H5端部分浏览器不支持 |
| loop | Boolean | 是否循环播放，默认 false | 否 |  |
| obeyMuteSwitch | Boolean | 是否遵循系统静音开关，当此参数为 false 时，即使用户打开了静音开关，也能继续发出声音，默认值 true | 否 | 微信小程序、百度小程序、头条小程序 |
| duration | Number | 当前音频的长度（单位：s），只有在当前有合法的 src 时返回 | 是 |  |
| currentTime | Number | 当前音频的播放位置（单位：s），只有在当前有合法的 src 时返回，时间不取整，保留小数点后 6 位 | 是 |  |
| paused | Boolean | 当前是是否暂停或停止状态，true 表示暂停或停止，false 表示正在播放 | 是 |  |
| buffered | Number | 音频缓冲的时间点，仅保证当前播放时间点到此时间点内容已缓冲。 | 是 |  |
| volume | Number | 音量。范围 0~1。 | 否 |  |

**innerAudioContext 对象的方法列表**

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| play |  | 播放（H5端部分浏览器需在用户交互时进行） |
| pause |  | 暂停 |
| stop |  | 停止 |
| seek | position | 跳转到指定位置，单位 s |
| destroy |  | 销毁当前实例 |
| onCanplay | callback | 音频进入可以播放状态，但不保证后面可以流畅播放 |
| onPlay | callback | 音频播放事件 |
| onPause | callback | 音频暂停事件 |
| onStop | callback | 音频停止事件 |
| onEnded | callback | 音频自然播放结束事件 |
| onTimeUpdate | callback | 音频播放进度更新事件 |
| onError | callback | 音频播放错误事件 |
| onWaiting | callback | 音频加载中事件，当音频因为数据不足，需要停下来加载时会触发 |
| onSeeking | callback | 音频进行 seek 操作事件 |
| onSeeked | callback | 音频完成 seek 操作事件 |
| offCanplay | callback | 取消监听 onCanplay 事件 |
| offPlay | callback | 取消监听 onPlay 事件 |
| offPause | callback | 取消监听 onPause 事件 |
| offStop | callback | 取消监听 onStop 事件 |
| offEnded | callback | 取消监听 onEnded 事件 |
| offTimeUpdate | callback | 取消监听 onTimeUpdate 事件 |
| offError | callback | 取消监听 onError 事件 |
| offWaiting | callback | 取消监听 onWaiting 事件 |
| offSeeking | callback | 取消监听 onSeeking 事件 |
| offSeeked | callback | 取消监听 onSeeked 事件 |

errCode 说明

| **errCode** | **说明** |
| --- | --- |
| 10001 | 系统错误 |
| 10002 | 网络错误 |
| 10003 | 文件错误 |
| 10004 | 格式错误 |
| -1 | 未知错误 |

**支持格式**

| **格式** | **iOS** | **Android** |
| --- | --- | --- |
| flac | x | √ |
| m4a | √ | √ |
| ogg | x | √ |
| ape | x | √ |
| amr | x | √ |
| wma | x | √ |
| wav | √ | √ |
| mp3 | √ | √ |
| mp4 | x | √ |
| aac | √ | √ |
| aiff | √ | x |
| caf | √ | x |

**示例**

const innerAudioContext = uni.createInnerAudioContext();

innerAudioContext.autoplay = true;

innerAudioContext.src = 'https://img-cdn-qiniu.dcloud.net.cn/uniapp/audio/music.mp3';

innerAudioContext.onPlay(() => {

console.log('开始播放');

});

innerAudioContext.onError((res) => {

console.log(res.errMsg);

console.log(res.errCode);

});

**tips**

如需音频的倍速播放，可通过video的倍速播放替代实现。插件市场有封装好的插件[音频倍速播放](https://ext.dcloud.net.cn/search?q=%E9%9F%B3%E9%A2%91%E5%80%8D%E9%80%9F%E6%92%AD%E6%94%BE)

## 视频

### [uni.chooseVideo(OBJECT)](https://uniapp.dcloud.io/api/media/video?id=choosevideo)

拍摄视频或从手机相册中选视频，返回视频的临时文件路径。另外选择和上传非图像、视频文件参考：<https://ask.dcloud.net.cn/article/35547>。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| sourceType | Array<String> | 否 | album 从相册选视频，camera 使用相机拍摄，默认为：['album', 'camera'] |  |
| compressed | Boolean | 否 | 是否压缩所选的视频源文件，默认值为 true，需要压缩。 | 微信小程序、百度小程序、头条小程序 |
| maxDuration | Number | 否 | 拍摄视频最长拍摄时间，单位秒。最长支持 60 秒。 | APP平台 1.9.7+(iOS支持，Android取决于ROM的拍照组件是否实现此功能，如果没实现此功能则忽略此属性。) 微信小程序、百度小程序 |
| camera | String | 否 | 'front'、'back'，默认'back' | APP、微信小程序 |
| success | Function | 否 | 接口调用成功，返回视频文件的临时文件路径，详见返回参数说明。 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**success 返回参数说明**

| **参数** | **说明** | **平台差异说明说明** |
| --- | --- | --- |
| tempFilePath | 选定视频的临时文件路径 |  |
| duration | 选定视频的时间长度，单位为 s | APP平台 2.1.0+、微信小程序 |
| size | 选定视频的数据量大小 | APP平台 2.1.0+、微信小程序 |
| height | 返回选定视频的高 | APP平台 2.1.0+、微信小程序 |
| width | 返回选定视频的宽 | APP平台 2.1.0+、微信小程序 |
| name | 包含扩展名的文件名称 | 仅H5支持 |

**注意：**

* 文件的临时路径，在应用本次启动期间可以正常使用，如需持久保存，需在主动调用 [uni.saveFile](https://uniapp.dcloud.io/api/file/file?id=savefile)，在应用下次启动时才能访问得到。
* camera 部分 Android 手机下由于系统 ROM 不支持无法生效，打开拍摄界面后可操作切换
* 可以通过用户授权API来判断用户是否给应用授予相册或摄像头的访问权限<https://uniapp.dcloud.io/api/other/authorize>
* App下如需进一步压缩视频大小，可以在插件市场搜索[视频压缩](http://ext.dcloud.net.cn/search?q=%E8%A7%86%E9%A2%91%E5%8E%8B%E7%BC%A9)插件

**示例**

<template>

<view>

<text>hello</text>

<button @tap="test">click me</button>

<video :src="src"></video>

</view>

</template>

export default {

data: {

src: ''

},

methods: {

test: function () {

var self = this;

uni.chooseVideo({

count: 1,

sourceType: ['camera', 'album'],

success: function (res) {

self.src = res.tempFilePath;

}

});

}

}

}

### [uni.saveVideoToPhotosAlbum(OBJECT)](https://uniapp.dcloud.io/api/media/video?id=savevideotophotosalbum)

保存视频到系统相册。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| filePath | String | 是 | 视频文件路径，可以是临时文件路径也可以是永久文件路径 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 调用结果 |

**注意**

* 可以通过用户授权API来判断用户是否给应用授予相册写入权限<https://uniapp.dcloud.io/api/other/authorize>

**示例**

<template>

<view>

<text>hello</text>

<button @tap="test">click me</button>

<video :src="src"></video>

</view>

</template>

export default {

data: {

src: ''

},

methods: {

test: function () {

var self = this;

uni.chooseVideo({

count: 1,

sourceType: ['camera'],

success: function (res) {

self.src = res.tempFilePath;

uni.saveVideoToPhotosAlbum({

filePath: res.tempFilePath,

success: function () {

console.log('save success');

}

});

}

});

}

}

}

## 视频组件控制

### [uni.createVideoContext(videoId, this)](https://uniapp.dcloud.io/api/media/video-context?id=createvideocontext)

创建并返回 video 上下文 videoContext 对象。在自定义组件下，第二个参数传入组件实例this，以操作组件内 <video> 组件。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | 基础库版本>=1.10.0 | √ | √ | √ |

**videoContext 对象的方法列表**

| **方法** | **参数** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| play | 无 | 播放 |  |
| pause | 无 | 暂停 |  |
| seek | position | 跳转到指定位置，单位 s |  |
| stop |  | 停止视频 | 微信小程序 |
| sendDanmu | danmu | 发送弹幕，danmu 包含两个属性 text, color |  |
| playbackRate | rate | 设置倍速播放，支持的倍率有 0.5/0.8/1.0/1.25/1.5。微信基础库2.6.3 起支持 2.0 倍速 |  |
| requestFullScreen | 无 | 进入全屏，可传入{direction}参数，详见 video 组件文档 |  |
| exitFullScreen | 无 | 退出全屏 |  |
| showStatusBar | 无 | 显示状态栏，仅在iOS全屏下有效 | 微信小程序、百度小程序、QQ小程序 |
| hideStatusBar | 无 | 隐藏状态栏，仅在iOS全屏下有效 | 微信小程序、百度小程序、QQ小程序 |

**注意：**

* app-nvue 平台 2.2.5+ 支持 uni.createVideoContext(videoId, this)
* app-nvue 平台 2.2.5以下使用本API，需同时设置组件属性id和ref <video id="video1" ref="video1"></video>，或者直接使用 ref，例如 this.$refs.video1

**示例**

<template>

<view>

<view class="page-body">

<view class="page-section">

<video id="myVideo" src="http://img.cdn.qiniu.dcloud.net.cn/wap2appvsnative.mp4" @error="videoErrorCallback" :danmu-list="danmuList"

enable-danmu danmu-btn controls></video>

<view class="uni-list">

<view class="uni-list-cell">

<view>

<view class="uni-label">弹幕内容</view>

</view>

<view class="uni-list-cell-db">

<input @blur="bindInputBlur" class="uni-input" type="text" placeholder="在此处输入弹幕内容" />

</view>

</view>

</view>

<view class="btn-area">

<button @tap="bindSendDanmu" class="page-body-button" formType="submit">发送弹幕</button>

</view>

</view>

</view>

</view>

</template>

export default {

data() {

return {

title: 'video',

src: '',

inputValue: '',

danmuList: [{

text: '第 1s 出现的弹幕',

color: '#ff0000',

time: 1

},

{

text: '第 3s 出现的弹幕',

color: '#ff00ff',

time: 3

}

]

}

},

onReady: function (res) {

this.videoContext = uni.createVideoContext('myVideo')

},

methods: {

bindInputBlur: function (e) {

this.inputValue = e.target.value

},

bindButtonTap: function () {

var that = this

uni.chooseVideo({

sourceType: ['album', 'camera'],

maxDuration: 60,

camera: ['front', 'back'],

success: function (res) {

this.src = res.tempFilePath

}

})

},

bindSendDanmu: function () {

this.videoContext.sendDanmu({

text: this.inputValue,

color: this.getRandomColor()

})

},

videoErrorCallback: function (e) {

console.log('视频错误信息:')

console.log(e.target.errMsg)

},

getRandomColor: function () {

const rgb = []

for (let i = 0; i < 3; ++i) {

let color = Math.floor(Math.random() \* 256).toString(16)

color = color.length == 1 ? '0' + color : color

rgb.push(color)

}

return '#' + rgb.join('')

}

}

}

## 相机组件控制

### [uni.createCameraContext()](https://uniapp.dcloud.io/api/media/camera-context?id=createcameracontext)

创建并返回 camera 组件的上下文 cameraContext 对象。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | √ |

本API为 camera 组件配套的js API，与 camera 组件的平台兼容性相同，可实现非全屏摄像头。App端可通过[plus.camera](https://www.html5plus.org/doc/zh_cn/camera.html" \t "_blank)实现全屏摄像头。

**cameraContext 对象的方法列表**

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| takePhoto | Object | 拍照，可指定质量，成功则返回图片路径。 |
| startRecord | Object | 开始录像 |
| stopRecord | Object | 结束录像，成功则返回封面与视频。 |
| onCameraFrame | Function | 获取 Camera 实时帧数据。仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/CameraContext.onCameraFrame.html) |

**takePhoto 的 Object 参数列表：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| quality | String | 否 | 成像质量，值为high（高质量）、normal（普通质量）、low（低质量），默认normal。 |
| success | Function | 否 | 接口调用成功的回调函数 ，返回照片文件的临时路径，res = { tempImagePath }。 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**startRecord 的 Object 参数列表：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| timeoutCallback | Function | 否 | 接超过30s或页面 onHide 时会结束录像 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**stopRecord 的 Object 参数列表：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调函数 ，返回封面与视频的临时路径，res = { tempThumbPath, tempVideoPath }。 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**注意**

* App下实现OCR等证件识别等需求，可在插件市场获取原生插件，<https://ext.dcloud.net.cn/plugin?id=135>
* 微信小程序下实现OCR等证件识别等需求，插件市场也有封装，搜索 [ocr](https://ext.dcloud.net.cn/search?q=ocr" \t "_blank) 可见。
* 可以通过用户授权API来判断用户是否给应用授予摄像头的访问权限<https://uniapp.dcloud.io/api/other/authorize>

## 直播组件控制

### [uni.createLivePlayerContext(livePlayerId, this)](https://uniapp.dcloud.io/api/media/live-player-context?id=createliveplayercontext)

创建 live-player 上下文 livePlayerContext 对象。注意是直播的播放而不是推流。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| 见下 | x | √ | x | √ | x | √ |

App平台的直播播放，不使用此API，而直接使用video的API。

**参数说明**

| **参数** | **说明** | **平台差异说明** |
| --- | --- | --- |
| livePlayerId | <live-player> 组件 id |  |
| this | 在自定义组件下，当前组件实例的 this，以操作组件内 <live-player> 组件 | 微信小程序 |

**livePlayerContext 对象的方法列表：**

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| play | Object | 播放 |
| stop | Object | 停止 |
| mute | Object | 静音 |
| pause | Object | 暂停 |
| resume | Object | 恢复 |
| requestFullScreen | Object | 进入全屏 |
| exitFullScreen | Object | 退出全屏 |

**requestFullScreen 的 Object 参数列表：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| direction | Number | 是 | 设置全屏时的方向，有效值为 0（正常竖向）, 90（屏幕逆时针90度）, -90（屏幕顺时针90度）。 |
| success | Function | 否 | 接口调用成功的回调函数。 |
| fail | Function | 否 | 接口调用失败的回调函数。 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行）。 |

**其他方法的 Object 参数列表：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [uni.createLivePusherContext(livePusherId, this)](https://uniapp.dcloud.io/api/media/live-player-context?id=createlivepushercontext)

创建 live-pusher 上下文 livePusherContext 对象。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| 见下 | x | √ | x | x | x | √ |

* app-nvue 平台 2.2.5+ 支持 uni.createLivePusherContext(livePusherId, this)
* app-nvue 平台 2.2.5以下，需要同时设置组件属性id和ref <live-pusher id="livepusher1" ref="livepusher1"></live-pusher>，或者直接使用 ref，例如 this.$refs.livepusher1
* app-vue 平台，需要编写条件编译代码，使用 plus.video.LivePusher，[业务指南](https://ask.dcloud.net.cn/article/13416" \t "_blank)、[规范文档](http://www.html5plus.org/doc/zh_cn/video.html#plus.video.LivePusher)

使用nvue做直播，比使用vue的优势有：

1. nvue可一套代码直接编译到App和微信
2. nvue的cover-view比vue的cover-view更强大，在视频上绘制元素更容易。如果只考虑App端的话，不用cover-view，任意组件都可以覆盖live-pusher组件
3. 若需要视频内嵌在swiper里上下滑动（类抖音、映客首页模式），App端只有nvue才能实现 当然nvue相比vue的坏处是css写法受限，如果只开发微信小程序，不考虑App，那么使用vue页面也是一样的。

**参数说明**

设置live-pusher组件的推流地址，推流视频模式等。

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| url | string |  | 是 | 推流地址，支持RTMP协议。 |
| mode | string |  | 否 | 推流视频模式，可取值：SD（标清）, HD（高清）, FHD（超清）。 |
| muted | Boolean | false | 否 | 是否静音。 |
| enable-camera | Boolean | true | 否 | 开启摄像头。 |
| auto-focus | Boolean | true | 否 | 自动聚集。 |
| beauty | Number | 0 | 否 | 美颜，取值范围 0-9（iOS取值范围为1） ，0 表示关闭。 |
| whiteness | Number | 0 | 否 | 美白，取值范围 0-9（iOS取值范围为1） ，0 表示关闭。 |

#### [API](https://uniapp.dcloud.io/api/media/live-player-context?id=api)

#### [start(callback)](https://uniapp.dcloud.io/api/media/live-player-context?id=start)

**开始推流**

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/media/live-player-context?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e)

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| type | String | "success" 表示成功， "fail" 表示失败 |

#### [pause(callback)](https://uniapp.dcloud.io/api/media/live-player-context?id=pause)

**暂停推流**

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/media/live-player-context?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e-1)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | String | "success" 表示成功， "fail" 表示失败 |

#### [resume(callback)](https://uniapp.dcloud.io/api/media/live-player-context?id=resume)

**恢复推流**

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/media/live-player-context?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e-2)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | String | "success" 表示成功， "fail" 表示失败 |

#### [stop(callback)](https://uniapp.dcloud.io/api/media/live-player-context?id=stop)

**停止推流**

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/media/live-player-context?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e-3)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | String | "success" 表示成功， "fail" 表示失败 |

#### [switchCamera(callback)](https://uniapp.dcloud.io/api/media/live-player-context?id=switchcamera)

**切换前后摄像头**

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/media/live-player-context?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e-4)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | String | "success" 表示成功， "fail" 表示失败 |

#### [snapshot(callback)](https://uniapp.dcloud.io/api/media/live-player-context?id=snapshot)

**快照**

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/media/live-player-context?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e-5)

##### [成功时的回调](https://uniapp.dcloud.io/api/media/live-player-context?id=%e6%88%90%e5%8a%9f%e6%97%b6%e7%9a%84%e5%9b%9e%e8%b0%83)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | string | "success" 表示成功, "fail" 表示失败 |
| code | Number | 对应code码 |
| message | object | {width:"快照图片宽度",height:"快照图片高度",tempImagePath:"快照图片路径"}。 |

##### [失败的回调](https://uniapp.dcloud.io/api/media/live-player-context?id=%e5%a4%b1%e8%b4%a5%e7%9a%84%e5%9b%9e%e8%b0%83)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | string | "fail" 表示失败 |
| code | Number |  |
| message | object |  |

#### [startPreview(callback)](https://uniapp.dcloud.io/api/media/live-player-context?id=startpreview)

**开启摄像头预览**

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/media/live-player-context?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e-6)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | String | "success" 表示成功， "fail" 表示失败 |

#### [stopPreview(callback)](https://uniapp.dcloud.io/api/media/live-player-context?id=stoppreview)

**关闭摄像头预览**

##### [callback 返回 Object 参数说明](https://uniapp.dcloud.io/api/media/live-player-context?id=callback-%e8%bf%94%e5%9b%9e-object-%e5%8f%82%e6%95%b0%e8%af%b4%e6%98%8e-7)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| type | String | "success" 表示成功， "fail" 表示失败 |

#### [事件](https://uniapp.dcloud.io/api/media/live-player-context?id=%e4%ba%8b%e4%bb%b6)

#### [statechange](https://uniapp.dcloud.io/api/media/live-player-context?id=statechange)

**状态变化事件**

##### [返回参数（detail）的详细说明](https://uniapp.dcloud.io/api/media/live-player-context?id=%e8%bf%94%e5%9b%9e%e5%8f%82%e6%95%b0%ef%bc%88detail%ef%bc%89%e7%9a%84%e8%af%a6%e7%bb%86%e8%af%b4%e6%98%8e)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| code | Number |  |
| message | string |  |

#### [netstatus](https://uniapp.dcloud.io/api/media/live-player-context?id=netstatus)

**网络状态通知事件**

##### [安卓 返回参数（detail）的详细说明](https://uniapp.dcloud.io/api/media/live-player-context?id=%e5%ae%89%e5%8d%93-%e8%bf%94%e5%9b%9e%e5%8f%82%e6%95%b0%ef%bc%88detail%ef%bc%89%e7%9a%84%e8%af%a6%e7%bb%86%e8%af%b4%e6%98%8e)

| **键名** | **说明** |
| --- | --- |
| videoBitrate | 当前视频编/码器输出的比特率，单位 kbps |
| audioBitrate | 当前音频编/码器输出的比特率，单位 kbps |
| videoFPS | 当前视频帧率 |
| videoGOP | 当前视频 GOP,也就是每两个关键帧(I帧)间隔时长，单位 s |
| netSpeed | 当前的发送/接收速度 |
| netJitter | 网络抖动情况，抖动越大，网络越不稳定 |
| videoWidth | 视频画面的宽度 |
| videoHeight | 视频画面的高度 |

##### [iOS 返回参数（detail）的详细说明](https://uniapp.dcloud.io/api/media/live-player-context?id=ios-%e8%bf%94%e5%9b%9e%e5%8f%82%e6%95%b0%ef%bc%88detail%ef%bc%89%e7%9a%84%e8%af%a6%e7%bb%86%e8%af%b4%e6%98%8e)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| code | Number | code码 |
| message | string | 具体的网络状态信息 |

#### [error](https://uniapp.dcloud.io/api/media/live-player-context?id=error)

**渲染错误事件**

##### [返回参数（detail）的详细说明](https://uniapp.dcloud.io/api/media/live-player-context?id=%e8%bf%94%e5%9b%9e%e5%8f%82%e6%95%b0%ef%bc%88detail%ef%bc%89%e7%9a%84%e8%af%a6%e7%bb%86%e8%af%b4%e6%98%8e-1)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| errCode | Number |  |
| errMsg | string |  |

## 富文本

### editorContext

editor 组件对应的 editorContext 实例，可通过 [uni.createSelectorQuery](https://uniapp.dcloud.io/api/ui/nodes-info?id=createselectorquery) 获取。

editorContext 通过 id 跟一个 [<editor>](https://uniapp.dcloud.io/component/editor) 组件绑定，操作对应的 [<editor>](https://uniapp.dcloud.io/component/editor) 组件。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | 2.4.5+ | √ | x | x | x | x |

### editorContext.format(name, value)

修改样式

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| name | String | 属性 |
| value | String | 值 |

**支持设置的样式列表**

| **name** | **value** |
| --- | --- |
| bold |  |
| italic |  |
| underline |  |
| strike |  |
| ins |  |
| script | sub / super |
| header | H1 / H2 / h3 / H4 / h5 / H6 |
| align | left / center / right / justify |
| direction | rtl |
| indent | -1 / +1 |
| list | ordered / bullet / check |
| color | hex color |
| backgroundColor | hex color |
| margin/marginTop/marginBottom/marginLeft/marginRight | css style |
| padding/paddingTop/paddingBottom/paddingLeft/paddingRight | css style |
| font/fontSize/fontStyle/fontVariant/fontWeight/fontFamily | css style |
| lineHeight | css style |
| letterSpacing | css style |
| textDecoration | css style |
| textIndent | css style |

对已经应用样式的选区设置会取消样式。css style 表示 css 中规定的允许值。

### editorContext.insertDivider(OBJECT)

插入分割线

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### editorContext.insertImage(OBJECT)

插入图片

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| src | String |  | 是 | 图片地址 |
| alt | String |  | 否 | 图像无法显示时的替代文本 |
| data | Object |  | 否 | data 被序列化为 name=value;name1=value2 的格式挂在属性 data-custom 上 |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### editorContext.insertText(OBJECT)

覆盖当前选区，设置一段文本

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| text | String |  | 否 | 文本内容 |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### editorContext.setContents(OBJECT)

初始化编辑器内容，hmlt和delta同时存在时仅delta生效

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| html | String |  | 否 | 带标签的HTML内容 |
| delta | Object |  | 否 | 表示内容的delta对象 |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### editorContext.getContents(OBJECT)

获取编辑器内容

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### editorContext.clear(OBJECT)

清空编辑器内容

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### editorContext.removeFormat(OBJECT)

清除当前选区的样式

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### editorContext.undo(OBJECT)

撤销

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### editorContext.redo(OBJECT)

恢复

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

# 设备

## 系统信息

### [uni.getSystemInfo(OBJECT)](https://uniapp.dcloud.io/api/system/info?id=getsysteminfo)

获取系统信息。

**OBJECT 参数说明：**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 是 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **参数** | **说明** | **平台差异说明** |
| --- | --- | --- |
| brand | 手机品牌 | App、微信小程序、百度小程序、头条小程序、QQ小程序 |
| model | 手机型号 |  |
| pixelRatio | 设备像素比 |  |
| screenWidth | 屏幕宽度 |  |
| screenHeight | 屏幕高度 |  |
| windowWidth | 可使用窗口宽度 |  |
| windowHeight | 可使用窗口高度 |  |
| windowTop | 可使用窗口的顶部位置 | App、H5 |
| windowBottom | 可使用窗口的底部位置 | App、H5 |
| statusBarHeight | 状态栏的高度 | 头条小程序不支持 |
| navigationBarHeight | 导航栏的高度 | 百度小程序 |
| titleBarHeight | 标题栏高度 | 支付宝小程序 |
| language | 应用设置的语言 | 头条小程序不支持 |
| version | 引擎版本号 | H5不支持 |
| storage | 设备磁盘容量 | 支付宝小程序 |
| currentBattery | 当前电量百分比 | 支付宝小程序 |
| appName | 宿主APP名称 | 头条小程序 |
| AppPlatform | App平台 | QQ小程序 |
| host | 宿主平台 | 百度小程序 |
| app | 当前运行的客户端 | 支付宝小程序 |
| cacheLocation | 上一次缓存的位置信息 | 百度小程序 |
| system | 操作系统版本 |  |
| platform | 客户端平台，值域为：ios、android |  |
| fontSizeSetting | 用户字体大小设置。以“我-设置-通用-字体大小”中的设置为准，单位：px | 微信小程序、支付宝小程序、百度小程序、QQ小程序 |
| SDKVersion | 客户端基础库版本 | 支付宝小程序和H5不支持 |
| swanNativeVersion | 宿主平台版本号 | 百度小程序 |
| albumAuthorized | 允许微信使用相册的开关（仅 iOS 有效） | 微信小程序 |
| cameraAuthorized | 允许微信使用摄像头的开关 | 微信小程序 |
| locationAuthorized | 允许微信使用定位的开关 | 微信小程序 |
| microphoneAuthorized | 允许微信使用麦克风的开关 | 微信小程序 |
| notificationAuthorized | 允许微信通知的开关 | 微信小程序 |
| notificationAlertAuthorized | 允许微信通知带有提醒的开关（仅 iOS 有效） | 微信小程序 |
| notificationBadgeAuthorized | 允许微信通知带有标记的开关（仅 iOS 有效） | 微信小程序 |
| notificationSoundAuthorized | 允许微信通知带有声音的开关（仅 iOS 有效） | 微信小程序 |
| bluetoothEnabled | 蓝牙的系统开关 | 微信小程序 |
| locationEnabled | 地理位置的系统开关 | 微信小程序 |
| wifiEnabled | Wi-Fi 的系统开关 | 微信小程序 |
| safeArea | 在竖屏正方向下的安全区域 | App、H5、微信小程序 |
| safeAreaInsets | 在竖屏正方向下的安全区域插入位置（2.5.3+） | App、H5、微信小程序 |

**Tips**

* 屏幕高度=状态栏高度+原生导航栏高度+可使用窗口高度+原生tabbar高度
* H5端，windowHeight不包含NavigationBar和TabBar的高度，windowTop等于NavigationBar高度，windowBottom等于TabBar高度，statusBarHeight为0
* APP端windowTop和windowBottom都是0，注意区分与H5端的差异

**safeArea 的结构**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| left | Number | 安全区域左上角横坐标 |
| right | Number | 安全区域右下角横坐标 |
| top | Number | 安全区域左上角纵坐标 |
| bottom | Number | 安全区域右下角纵坐标 |
| width | Number | 安全区域的宽度，单位逻辑像素 |
| height | Number | 安全区域的高度，单位逻辑像素 |

**safeAreaInsets 的结构**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| left | Number | 安全区域左侧插入位置 |
| right | Number | 安全区域右侧插入位置 |
| top | Number | 安全区顶部插入位置 |
| bottom | Number | 安全区域底部插入位置 |

**示例**

uni.getSystemInfo({

success: function (res) {

console.log(res.model);

console.log(res.pixelRatio);

console.log(res.windowWidth);

console.log(res.windowHeight);

console.log(res.language);

console.log(res.version);

console.log(res.platform);

}

});

### [uni.getSystemInfoSync()](https://uniapp.dcloud.io/api/system/info?id=getsysteminfosync)

获取系统信息同步接口。

**同步返回参数说明**

| **参数** | **说明** | **平台差异说明** |
| --- | --- | --- |
| brand | 手机品牌 | App、微信小程序、百度小程序、头条小程序、QQ小程序 |
| model | 手机型号 |  |
| pixelRatio | 设备像素比 |  |
| screenWidth | 屏幕宽度 |  |
| screenHeight | 屏幕高度 |  |
| windowWidth | 可使用窗口宽度 |  |
| windowHeight | 可使用窗口高度 |  |
| windowTop | 可使用窗口的顶部位置 | App、H5 |
| windowBottom | 可使用窗口的底部位置 | App、H5 |
| statusBarHeight | 状态栏的高度 | 头条小程序不支持 |
| navigationBarHeight | 导航栏的高度 | 百度小程序 |
| titleBarHeight | 标题栏高度 | 支付宝小程序 |
| language | 应用设置的语言 | 头条小程序不支持 |
| version | 引擎版本号 | H5不支持 |
| storage | 设备磁盘容量 | 支付宝小程序 |
| currentBattery | 当前电量百分比 | 支付宝小程序 |
| appName | 宿主APP名称 | 头条小程序 |
| AppPlatform | App平台 | QQ小程序 |
| host | 宿主平台 | 百度小程序 |
| app | 当前运行的客户端 | 支付宝小程序 |
| cacheLocation | 上一次缓存的位置信息 | 百度小程序 |
| system | 操作系统版本 |  |
| platform | 客户端平台，值域为：ios、android |  |
| fontSizeSetting | 用户字体大小设置。以“我-设置-通用-字体大小”中的设置为准，单位：px | 微信小程序、支付宝小程序、百度小程序、QQ小程序 |
| SDKVersion | 客户端基础库版本 | 支付宝小程序和H5不支持 |
| swanNativeVersion | 宿主平台版本号 | 百度小程序 |
| albumAuthorized | 允许微信使用相册的开关（仅 iOS 有效） | 微信小程序 |
| cameraAuthorized | 允许微信使用摄像头的开关 | 微信小程序 |
| locationAuthorized | 允许微信使用定位的开关 | 微信小程序 |
| microphoneAuthorized | 允许微信使用麦克风的开关 | 微信小程序 |
| notificationAuthorized | 允许微信通知的开关 | 微信小程序 |
| notificationAlertAuthorized | 允许微信通知带有提醒的开关（仅 iOS 有效） | 微信小程序 |
| notificationBadgeAuthorized | 允许微信通知带有标记的开关（仅 iOS 有效） | 微信小程序 |
| notificationSoundAuthorized | 允许微信通知带有声音的开关（仅 iOS 有效） | 微信小程序 |
| bluetoothEnabled | 蓝牙的系统开关 | 微信小程序 |
| locationEnabled | 地理位置的系统开关 | 微信小程序 |
| wifiEnabled | Wi-Fi 的系统开关 | 微信小程序 |
| safeArea | 在竖屏正方向下的安全区域 | App、H5、微信小程序 |
| safeAreaInsets | 在竖屏正方向下的安全区域插入位置（2.5.3+） | App、H5、微信小程序 |

**Tips**

* 使用注意同上getSystemInfo

**safeArea 的结构**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| left | Number | 安全区域左上角横坐标 |
| right | Number | 安全区域右下角横坐标 |
| top | Number | 安全区域左上角纵坐标 |
| bottom | Number | 安全区域右下角纵坐标 |
| width | Number | 安全区域的宽度，单位逻辑像素 |
| height | Number | 安全区域的高度，单位逻辑像素 |

**safeAreaInsets 的结构**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| left | Number | 安全区域左侧插入位置 |
| right | Number | 安全区域右侧插入位置 |
| top | Number | 安全区顶部插入位置 |
| bottom | Number | 安全区域底部插入位置 |

**示例**

try {

const res = uni.getSystemInfoSync();

console.log(res.model);

console.log(res.pixelRatio);

console.log(res.windowWidth);

console.log(res.windowHeight);

console.log(res.language);

console.log(res.version);

console.log(res.platform);

} catch (e) {

// error

}

**获取设备标记**

H5、小程序、iOS，属于对用户隐私保护比较严格的平台，在这些平台很难获取有效的设备唯一标记。

Android已经改进用户隐私保护，在很多新手机上，获取imei等信息时需要弹框让用户授权。而Android10已经无法获取imei了。

* H5平台： 常用的方式是uv，即在uni.storage里存一个随机数，本质是存在浏览器的localstorage里。将随机数发给服务器，进行用户身份识别和统计。当然如果用户浏览器清空了localstorage、更换了浏览器、或使用隐私模式，那么就统计数据就会有误差。
* 小程序平台： 小程序也可以采用与H5类似的方式，在uni.storage里存一个随机数。如果想获取用户的微信唯一ID，也可以弹框请求用户授权。
* App iOS平台： iOS并不提供imei的获取API，可通过[plus.device.getInfo](http://www.html5plus.org/doc/zh_cn/device.html#plus.device.getInfo) 可以获得设备的唯一标识（uuid），这个id其实也是一种随机数概念，类似于h5的uv计算，卸载app重装会发生变化；

iOS还有一个叫idfa的广告识别符，可通过Native.js获取，详见：[idfa介绍](https://ask.dcloud.net.cn/article/36107)。

* App Android平台： Android也可以使用UUID，同iOS。

Android10以下可以得到imei，在[plus.device.getInfo](http://www.html5plus.org/doc/zh_cn/device.html#plus.device.getInfo) 可以获得设备的国际移动设备身份码（imei）。注意很多新手机在获取imei时会弹框要求用户授权。

Android10以上，部分国产手机支持OAID，详见[匿名设备标识符（OAID）](http://www.html5plus.org/doc/zh_cn/device.html#plus.device.getOAID)

plus.device.getInfo的API是从HBuilderX 2.0.3+开始提供的，老版需使用plus.devide.uuid或plus.device.imei。

### [uni.canIUse(String)](https://uniapp.dcloud.io/api/system/info?id=caniuse)

判断应用的 API，回调，参数，组件等是否在当前版本可用。

平台差异说明

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**String 参数说明**

使用 ${API}.${method}.${param}.${options} 或者 ${component}.${attribute}.${option} 方式来调用，例如：

* ${API} 代表 API 名字
* ${method} 代表调用方式，有效值为return, success, object, callback
* ${param} 代表参数或者返回值
* ${options} 代表参数的可选值
* ${component} 代表组件名字
* ${attribute} 代表组件属性
* ${option} 代表组件属性的可选值

**示例**

uni.canIUse('getSystemInfoSync.return.screenWidth');

uni.canIUse('getSystemInfo.success.screenWidth');

uni.canIUse('showToast.object.image');

uni.canIUse('request.object.method.GET');

uni.canIUse('live-player');

uni.canIUse('text.selectable');

uni.canIUse('button.open-type.contact');

## 内存

### [uni.onMemoryWarning(CALLBACK)](https://uniapp.dcloud.io/api/system/memory?id=onmemorywarning)

监听内存不足告警事件。

当 iOS/Android 向小程序进程发出内存警告时，触发该事件。Android 下有告警等级划分，iOS 无等级划分。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | √ | x | √ |

**CALLBACK返回参数：**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| level | Number | 仅 Android 有该字段，对应系统内存告警等级宏定义 |

**level 的合法值**

| **值** | **说明** |
| --- | --- |
| 5 | TRIM\_MEMORY\_RUNNING\_MODERATE |
| 10 | TRIM\_MEMORY\_RUNNING\_LOW |
| 15 | TRIM\_MEMORY\_RUNNING\_CRITICAL |

**代码示例**

uni.onMemoryWarning(function () {

console.log('onMemoryWarningReceive')

})

## 网络状态

### [uni.getNetworkType(OBJECT)](https://uniapp.dcloud.io/api/system/network?id=getnetworktype)

获取网络类型。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 是 | 接口调用成功，返回网络类型 networkType |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数** | **说明** |
| --- | --- |
| networkType | 网络类型 |

**networkType 有效值**

| **值** | **说明** | **平台差异说明** |
| --- | --- | --- |
| wifi | wifi 网络 |  |
| 2g | 2g 网络 |  |
| 3g | 3g 网络 |  |
| 4g | 4g 网络 |  |
| ethernet | 有线网络 | App |
| unknown | Android 下不常见的网络类型 |  |
| none | 无网络 |  |

**示例**

uni.getNetworkType({

success: function (res) {

console.log(res.networkType);

}

});

### [uni.onNetworkStatusChange(CALLBACK)](https://uniapp.dcloud.io/api/system/network?id=onnetworkstatuschange)

监听网络状态变化。

**CALLBACK 返回参数**

| **参数** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| isConnected | Boolean | 当前是否有网络连接 | 头条小程序不支持 |
| networkType | String | 网络类型 |  |

**示例**

uni.onNetworkStatusChange(function (res) {

console.log(res.isConnected);

console.log(res.networkType);

});

## 加速度计

### [uni.onAccelerometerChange(CALLBACK)](https://uniapp.dcloud.io/api/system/accelerometer?id=onaccelerometerchange)

监听加速度数据，频率：5次/秒，接口调用后会自动开始监听，可使用 uni.stopAccelerometer 停止监听。

**CALLBACK 返回参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| x | Number | X 轴 |
| y | Number | Y 轴 |
| z | Number | Z 轴 |

**Tips**

* H5端获取加速度信息，需要部署在 **https** 服务上，本地预览（localhost）仍然可以使用 http 协议。

**示例**

uni.onAccelerometerChange(function (res) {

console.log(res.x);

console.log(res.y);

console.log(res.z);

});

### [uni.startAccelerometer(OBJECT)](https://uniapp.dcloud.io/api/system/accelerometer?id=startaccelerometer)

开始监听加速度数据。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| √ | √ | √ | x | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **默认** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| interval | String | normal | 否 | 接口调用成功的回调 | 微信小程序、百度小程序 |
| success | Function |  | 否 | 接口调用成功的回调 |  |
| fail | Function |  | 否 | 接口调用失败的回调函数 |  |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

interval 的合法值

| **值** | **说明** |
| --- | --- |
| game | 适用于更新游戏的回调频率，在 20ms/次 左右 |
| ui | 适用于更新 UI 的回调频率，在 60ms/次 左右 |
| normal | 普通的回调频率，在 200ms/次 左右 |

**示例**

uni.startAccelerometer();

### [uni.stopAccelerometer(OBJECT)](https://uniapp.dcloud.io/api/system/accelerometer?id=stopaccelerometer)

停止监听加速度数据。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.stopAccelerometer();

## 罗盘

### [uni.onCompassChange(CALLBACK)](https://uniapp.dcloud.io/api/system/compass?id=oncompasschange)

监听罗盘数据，频率：5次/秒，接口调用后会自动开始监听，可使用 uni.stopCompass 停止监听。

**CALLBACK 返回参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| direction | Number | 面对的方向度数 |

**Tips**

* H5端获取罗盘信息，需要部署在 **https** 服务上，本地预览（localhost）仍然可以使用 http 协议。

**示例**

uni.onCompassChange(function (res) {

console.log(res.direction);

});

### [uni.startCompass(OBJECT)](https://uniapp.dcloud.io/api/system/compass?id=startcompass)

开始监听罗盘数据。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | x | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.startCompass();

### [uni.stopCompass(OBJECT)](https://uniapp.dcloud.io/api/system/compass?id=stopcompass)

停止监听罗盘数据。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.stopCompass();

## 陀螺仪

### [uni.onGyroscopeChange(CALLBACK)](https://uniapp.dcloud.io/api/system/gyroscope?id=ongyroscopechange)

监听陀螺仪数据变化事件。

支付宝小程序频率为 500ms/次，微信小程序频率根据 [uni.startGyroscope](https://uniapp.dcloud.io/api/system/gyroscope?id=startgyroscope) 的 interval 参数设置。事件只有在调用 [uni.startGyroscope](https://uniapp.dcloud.io/api/system/gyroscope?id=startgyroscope) 后才会开始监听，使用 [uni.stopGyroscope](https://uniapp.dcloud.io/api/system/gyroscope?id=stopgyroscope) 可以停止监听。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | x | x |

**CALLBACK 参数说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| res | Object | res = {x,y,x} |

**res 的结构**

| **名称** | **类型** | **描述** |
| --- | --- | --- |
| x | number | x轴方向角速度 |
| y | number | y轴方向角速度 |
| z | number | z轴方向角速度 |

### [uni.startGyroscope(OBJECT)](https://uniapp.dcloud.io/api/system/gyroscope?id=startgyroscope)

开始监听陀螺仪数据。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | x | x |

| **属性** | **类型** | **默认值** | **必填** | **说明** | **平台说明** |
| --- | --- | --- | --- | --- | --- |
| interval | string | normal | 否 | 监听陀螺仪数据回调函数的执行频率：game（20ms/次）、ui（60ms/次）、normal（200ms/次） | 微信小程序 |
| success | function |  | 否 | 接口调用成功的回调函数 |  |
| fail | function |  | 否 | 接口调用失败的回调函数 |  |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

### [uni.stopGyroscope(OBJECT)](https://uniapp.dcloud.io/api/system/gyroscope?id=stopgyroscope)

停止监听陀螺仪数据。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | x | x |

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | function | 否 | 接口调用成功的回调函数 |
| fail | function | 否 | 接口调用失败的回调函数 |
| complete | function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**代码示例**

<template>

<view>

<button @click="start">开始监听陀螺仪变化</button>

<button @click="stop">停止监听陀螺仪变化</button>

</view>

</template>

export default {

methods: {

start() {

uni.onGyroscopeChange((res) => {

console.log('gyroData.rotationRate.x = ' + res.x)

console.log('gyroData.rotationRate.y = ' + res.y)

console.log('gyroData.rotationRate.z = ' + res.z)

});

uni.startGyroscope({

interval: "normal",

success() {

console.log('success')

},

fail() {

console.log('fail')

}

})

},

stop(){

uni.stopGyroscope({

success() {

console.log('stop success!')

},

fail() {

console.log('stop fail!')

}

})

}

}

}

**Tips**

* 陀螺仪 相关 API 在小程序开发工具中调用可能报错，请在真机中测试

## 拨打电话

### [uni.makePhoneCall(OBJECT)](https://uniapp.dcloud.io/api/system/phone?id=makephonecall)

拨打电话。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| phoneNumber | String | 是 | 需要拨打的电话号码 |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.makePhoneCall({

phoneNumber: '114' //仅为示例

});

注：App端关于电话短信的扩展文档

* Android不弹出询问框直接拨打电话：<https://ask.dcloud.net.cn/question/4035>
* 发送短信：<http://www.html5plus.org/doc/zh_cn/messaging.html>
* Android读取短信验证码：<http://ask.dcloud.net.cn/article/676>
* Android遍历读取短信：<https://ask.dcloud.net.cn/article/12934> 注意需要赋予相关权限。

## 扫码

### [uni.scanCode(OBJECT)](https://uniapp.dcloud.io/api/system/barcode?id=scancode)

调起客户端扫码界面，扫码成功后返回对应的结果。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| onlyFromCamera | Boolean | 否 | 是否只能从相机扫码，不允许从相册选择图片 | 头条小程序不支持此参数 |
| scanType | Array | 否 | 扫码类型，参数类型是数组，二维码是'qrCode'，一维码是'barCode'，DataMatrix是‘datamatrix’，pdf417是‘pdf417’。 | 头条小程序不支持此参数，支付宝只支持条码和二维码 |
| success | Function | 否 | 接口调用成功的回调，返回内容详见返回参数说明。 |  |
| fail | Function | 否 | 接口调用失败的回调函数（识别失败、用户取消等情况下触发） |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**success 返回参数说明**

| **参数** | **说明** | **平台差异说明** |
| --- | --- | --- |
| result | 所扫码的内容 |  |
| scanType | 所扫码的类型 | App、微信小程序、百度小程序、QQ小程序 |
| charSet | 所扫码的字符集 | App、微信小程序、百度小程序、QQ小程序 |
| path | 当所扫的码为当前应用的合法二维码时，会返回此字段，内容为二维码携带的 path。 | App、微信小程序、百度小程序、QQ小程序 |

**示例**

// 允许从相机和相册扫码

uni.scanCode({

success: function (res) {

console.log('条码类型：' + res.scanType);

console.log('条码内容：' + res.result);

}

});

// 只允许通过相机扫码

uni.scanCode({

onlyFromCamera: true,

success: function (res) {

console.log('条码类型：' + res.scanType);

console.log('条码内容：' + res.result);

}

});

// 调起条码扫描

uni.scanCode({

scanType: 'barCode',

success: function (res) {

console.log('条码类型：' + res.scanType);

console.log('条码内容：' + res.result);

}

});

**Tip**

* App-vue如果想自定义扫码，可参考[uni-app中如何使用5+的原生界面控件](http://ask.dcloud.net.cn/article/35036)和[plus.barcode API](https://www.html5plus.org/doc/zh_cn/barcode.html)
* App-nvue，支持barcode组件，可自定义扫码界面。[详见](https://uniapp.dcloud.io/component/barcode)。App端自定义扫码界面，建议使用nvue方式。
* 微信小程序自定义扫码界面，可使用camera组件。[详见](https://uniapp.dcloud.io/component/camera)
* 微信内嵌浏览器运行H5版时，可通过js sdk实现扫码，需要引入一个单独的js，[详见](https://ask.dcloud.net.cn/article/35380" \t "_blank)
* 在扫码界面点击返回也会进入 fail 回调中
* 支付宝小程序不支持 success 回调中的 scanType，charSet，path

## 剪切板

**剪贴板 API 平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

### [uni.setClipboardData(OBJECT)](https://uniapp.dcloud.io/api/system/clipboard?id=setclipboarddata)

设置系统剪贴板的内容。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| data | String | 是 | 需要设置的内容 |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.setClipboardData({

data: 'hello',

success: function () {

console.log('success');

}

});

### [uni.getClipboardData(OBJECT)](https://uniapp.dcloud.io/api/system/clipboard?id=getclipboarddata)

获取系统剪贴板内容。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| data | String | 剪贴板的内容 |

**示例**

uni.getClipboardData({

success: function (res) {

console.log(res.data);

}

});

#### [****注意****](https://uniapp.dcloud.io/api/system/clipboard?id=%e6%b3%a8%e6%84%8f)

* 设置剪贴板内容后，小程序平台会自动弹出轻提示。App平台默认与小程序保持一致策略。如不希望在App平台弹出提示，可使用Native.js自行操作剪贴板，插件市场有封装好的示例<https://ext.dcloud.net.cn/plugin?id=712>。也可以在设置剪切板后立即uni.hideToast()。
* H5的复制粘贴，可去插件市场搜索[剪贴板](https://ext.dcloud.net.cn/search?q=%E5%89%AA%E8%B4%B4%E6%9D%BF)

## 屏幕

### [uni.setScreenBrightness(OBJECT)](https://uniapp.dcloud.io/api/system/brightness?id=setscreenbrightness)

设置屏幕亮度。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | x | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| value | Number | 是 | 屏幕亮度值，范围 0~1，0 最暗，1 最亮 |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.setScreenBrightness({

value: 0.5,

success: function () {

console.log('success');

}

});

### [uni.getScreenBrightness(OBJECT)](https://uniapp.dcloud.io/api/system/brightness?id=getscreenbrightness)

获取屏幕亮度。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | x | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| value | Number | 屏幕亮度值，范围 0~1，0 最暗，1 最亮。 |

**示例**

uni.getScreenBrightness({

success: function (res) {

console.log('屏幕亮度值：' + res.value);

}

});

### [uni.setKeepScreenOn(OBJECT)](https://uniapp.dcloud.io/api/system/brightness?id=setkeepscreenon)

设置是否保持常亮状态。仅在当前应用生效，离开应用后设置失效。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| keepScreenOn | Boolean | 是 | 接口调用成功的回调 |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 调用结果 |

**示例**

// 保持屏幕常亮

uni.setKeepScreenOn({

keepScreenOn: true

});

## 用户截屏事件

### [uni.onUserCaptureScreen(CALLBACK)](https://uniapp.dcloud.io/api/system/capture-screen?id=onusercapturescreen)

监听用户主动截屏事件，用户使用系统截屏按键截屏时触发此事件。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | √ | √ | √ |

**CALLBACK返回参数：**

无

**代码示例**

uni.onUserCaptureScreen(function() {

console.log('用户截屏了')

});

## 振动

### [uni.vibrate(OBJECT)](https://uniapp.dcloud.io/api/system/vibrate?id=vibrate)

使手机发生振动。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.vibrate({

success: function () {

console.log('success');

}

});

### [uni.vibrateLong(OBJECT)](https://uniapp.dcloud.io/api/system/vibrate?id=vibratelong)

使手机发生较长时间的振动（400ms）。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.vibrateLong({

success: function () {

console.log('success');

}

});

### [uni.vibrateShort(OBJECT)](https://uniapp.dcloud.io/api/system/vibrate?id=vibrateshort)

使手机发生较短时间的振动（15ms）。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.vibrateShort({

success: function () {

console.log('success');

}

});

**注意**

* iOS上只有长震动，没有短震动
* iOS上需要手机设置“打开响铃时震动”或“静音时震动”，否则无法震动
* vibrate只适用于钉钉小程序、支付宝小程序

## 手机联系人

### [uni.addPhoneContact(OBJECT)](https://uniapp.dcloud.io/api/system/contact?id=addphonecontact)

调用后，用户可以选择将该表单以“新增联系人”或“添加到已有联系人”的方式（APP端目前没有选择步骤，将直接写入），写入手机系统通讯录，完成手机通讯录联系人和联系方式的增加。

App平台提供了更多通讯录相关API，包括读取联系人，详见：<https://www.html5plus.org/doc/zh_cn/contacts.html>

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | x | x |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| photoFilePath | String | 否 | 头像本地文件路径 |
| nickName | String | 否 | 昵称 |
| lastName | String | 否 | 姓氏 |
| middleName | String | 否 | 中间名 |
| firstName | String | 是 | 名字 |
| remark | String | 否 | 备注 |
| mobilePhoneNumber | String | 否 | 手机号 |
| weChatNumber | String | 否 | 微信号 |
| addressCountry | String | 否 | 联系地址国家 |
| addressState | String | 否 | 联系地址省份 |
| addressCity | String | 否 | 联系地址城市 |
| addressStreet | String | 否 | 联系地址街道 |
| addressPostalCode | String | 否 | 联系地址邮政编码 |
| organization | String | 否 | 公司 |
| title | String | 否 | 职位 |
| workFaxNumber | String | 否 | 工作传真 |
| workPhoneNumber | String | 否 | 工作电话 |
| hostNumber | String | 否 | 公司电话 |
| email | String | 否 | 电子邮件 |
| url | String | 否 | 网站 |
| workAddressCountry | String | 否 | 工作地址国家 |
| workAddressState | String | 否 | 工作地址省份 |
| workAddressCity | String | 否 | 工作地址城市 |
| workAddressStreet | String | 否 | 工作地址街道 |
| workAddressPostalCode | String | 否 | 工作地址邮政编码 |
| homeFaxNumber | String | 否 | 住宅传真 |
| homePhoneNumber | String | 否 | 住宅电话 |
| homeAddressCountry | String | 否 | 住宅地址国家 |
| homeAddressState | String | 否 | 住宅地址省份 |
| homeAddressCity | String | 否 | 住宅地址城市 |
| homeAddressStreet | String | 否 | 住宅地址街道 |
| homeAddressPostalCode | String | 否 | 住宅地址邮政编码 |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**回调结果**

| **回调类型** | **errMsg** | **说明** |
| --- | --- | --- |
| success | ok | 添加成功 |
| cancel | fail cancel | 用户取消操作 |
| fail | fail ${detail} | 调用失败，detail 加上详细信息。 |

**示例**

uni.addPhoneContact({

nickName: '昵称',

lastName: '姓',

firstName: '名',

remark: '备注',

mobilePhoneNumber: '114',

weChatNumber: 'wx123',

success: function () {

console.log('success');

},

fail: function () {

console.log('fail');

}

});

**注意**

* 手机OS对通讯录访问有严格的权限限制和要求。在小程序中使用时，需注意微信等小程序载体本身已经获得了手机端的授权许可。App端获取通讯录相关权限，参考<https://ext.dcloud.net.cn/plugin?id=594>
* 打包App时，云打包则需要在manifest中配置权限和模块，离线打包需自行在原生工程中配置。

## 蓝牙

**蓝牙 API 平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | x | x | x |

### [uni.openBluetoothAdapter(OBJECT)](https://uniapp.dcloud.io/api/system/bluetooth?id=openbluetoothadapter)

初始化蓝牙模块

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/bluetooth?id=%e9%94%99%e8%af%af)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

**注意**

* 其他蓝牙相关 API 必须在 [uni.openBluetoothAdapter](https://uniapp.dcloud.io/api/system/bluetooth?id=openbluetoothadapter) 调用之后使用。否则 API 会返回错误（errCode=10000）。
* 在用户蓝牙开关未开启或者手机不支持蓝牙功能的情况下，调用 [uni.openBluetoothAdapter](https://uniapp.dcloud.io/api/system/bluetooth?id=openbluetoothadapter) 会返回错误（errCode=10001），表示手机蓝牙功能不可用。此时APP蓝牙模块已经初始化完成，可通过 [uni.onBluetoothAdapterStateChange](https://uniapp.dcloud.io/api/system/bluetooth?id=onbluetoothadapterstatechange) 监听手机蓝牙状态的改变，也可以调用蓝牙模块的所有API。

**示例代码**

uni.openBluetoothAdapter({

success(res) {

console.log(res)

}

})

### [uni.startBluetoothDevicesDiscovery(OBJECT)](https://uniapp.dcloud.io/api/system/bluetooth?id=startbluetoothdevicesdiscovery)

开始搜寻附近的蓝牙外围设备。**此操作比较耗费系统资源，请在搜索并连接到设备后调用**[uni.stopBluetoothDevicesDiscovery](https://uniapp.dcloud.io/api/system/bluetooth?id=stopbluetoothdevicesdiscovery)**方法停止搜索。**

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| services | Array<String> |  | 否 | 要搜索但蓝牙设备主 service 的 uuid 列表。某些蓝牙设备会广播自己的主 service 的 uuid。如果设置此参数，则只搜索广播包有对应 uuid 的主服务的蓝牙设备。建议主要通过该参数过滤掉周边不需要处理的其他蓝牙设备。 |
| allowDuplicatesKey | boolean | false | 否 | 是否允许重复上报同一设备。如果允许重复上报，则 uni.onBlueToothDeviceFound 方法会多次上报同一设备，但是 RSSI 值会有不同。 |
| interval | number | 0 | 否 | 上报设备的间隔。0 表示找到新设备立即上报，其他数值根据传入的间隔上报。 |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/bluetooth?id=%e9%94%99%e8%af%af-1)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

**注意：**

* App 端目前仅支持发现ble蓝牙设备，更多蓝牙设备发现需使用 Native.js，参考：<https://ask.dcloud.net.cn/article/114>。

**示例代码**

// 以微信硬件平台的蓝牙智能灯为例，主服务的 UUID 是 FEE7。传入这个参数，只搜索主服务 UUID 为 FEE7 的设备

uni.startBluetoothDevicesDiscovery({

services: ['FEE7'],

success(res) {

console.log(res)

}

})

### [uni.onBluetoothDeviceFound(CALLBACK)](https://uniapp.dcloud.io/api/system/bluetooth?id=onbluetoothdevicefound)

监听寻找到新设备的事件

**CALLBACK 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| devices | Array<Object> | 新搜索到的设备列表 |

**devices 的结构**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| name | string | 蓝牙设备名称，某些设备可能没有 |
| deviceId | string | 用于区分设备的 id |
| RSSI | number | 当前蓝牙设备的信号强度 |
| advertisData | ArrayBuffer | 当前蓝牙设备的广播数据段中的 ManufacturerData 数据段。 |
| advertisServiceUUIDs | Array<String> | 当前蓝牙设备的广播数据段中的 ServiceUUIDs 数据段 |
| localName | string | 当前蓝牙设备的广播数据段中的 LocalName 数据段 |
| serviceData | Object | 当前蓝牙设备的广播数据段中的 ServiceData 数据段 |

**注意**

* 若在 [uni.onBluetoothDeviceFound](https://uniapp.dcloud.io/api/system/bluetooth?id=onbluetoothdevicefound) 回调了某个设备，则此设备会添加到 [uni.getBluetoothDevices](https://uniapp.dcloud.io/api/system/bluetooth?id=getbluetoothdevices) 接口获取到的数组中。

**示例代码**

// ArrayBuffer转16进度字符串示例

function ab2hex(buffer) {

const hexArr = Array.prototype.map.call(

new Uint8Array(buffer),

function (bit) {

return ('00' + bit.toString(16)).slice(-2)

}

)

return hexArr.join('')

}

uni.onBluetoothDeviceFound(function (devices) {

console.log('new device list has founded')

console.dir(devices)

console.log(ab2hex(devices[0].advertisData))

})

### [uni.stopBluetoothDevicesDiscovery(OBJECT)](https://uniapp.dcloud.io/api/system/bluetooth?id=stopbluetoothdevicesdiscovery)

停止搜寻附近的蓝牙外围设备。若已经找到需要的蓝牙设备并不需要继续搜索时，建议调用该接口停止蓝牙搜索。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/bluetooth?id=%e9%94%99%e8%af%af-2)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

**示例代码**

uni.stopBluetoothDevicesDiscovery({

success(res) {

console.log(res)

}

})

### [uni.onBluetoothAdapterStateChange(CALLBACK)](https://uniapp.dcloud.io/api/system/bluetooth?id=onbluetoothadapterstatechange)

监听蓝牙适配器状态变化事件

**CALLBACK 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| available | boolean | 蓝牙适配器是否可用 |
| discovering | boolean | 蓝牙适配器是否处于搜索状态 |

**示例代码**

uni.onBluetoothAdapterStateChange(function (res) {

console.log('adapterState changed, now is', res)

})

### [uni.getConnectedBluetoothDevices(OBJECT)](https://uniapp.dcloud.io/api/system/bluetooth?id=getconnectedbluetoothdevices)

根据 uuid 获取处于已连接状态的设备。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| services | Array<String> |  | 是 | 蓝牙设备主 service 的 uuid 列表 |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| devices | Array<Object> | 搜索到的设备列表 |

**res.devices 的结构**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| name | string | 蓝牙设备名称，某些设备可能没有 |
| deviceId | string | 用于区分设备的 id |

#### [错误](https://uniapp.dcloud.io/api/system/bluetooth?id=%e9%94%99%e8%af%af-3)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

**示例代码**

uni.getConnectedBluetoothDevices({

success(res) {

console.log(res)

}

})

### [uni.getBluetoothDevices(OBJECT)](https://uniapp.dcloud.io/api/system/bluetooth?id=getbluetoothdevices)

获取在蓝牙模块生效期间所有已发现的蓝牙设备。包括已经和本机处于连接状态的设备。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| devices | Array<Object> | uuid 对应的的已连接设备列表 |

**res.devices 的结构**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| name | string | 蓝牙设备名称，某些设备可能没有 |
| deviceId | string | 用于区分设备的 id |
| RSSI | number | 当前蓝牙设备的信号强度 |
| advertisData | ArrayBuffer | 当前蓝牙设备的广播数据段中的 ManufacturerData 数据段。 |
| advertisServiceUUIDs | Array<String> | 当前蓝牙设备的广播数据段中的 ServiceUUIDs 数据段 |
| localName | string | 当前蓝牙设备的广播数据段中的 LocalName 数据段 |
| serviceData | Object | 当前蓝牙设备的广播数据段中的 ServiceData 数据段 |

#### [错误](https://uniapp.dcloud.io/api/system/bluetooth?id=%e9%94%99%e8%af%af-4)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

**示例代码**

// ArrayBuffer转16进度字符串示例

function ab2hex(buffer) {

const hexArr = Array.prototype.map.call(

new Uint8Array(buffer),

function (bit) {

return ('00' + bit.toString(16)).slice(-2)

}

)

return hexArr.join('')

}

uni.getBluetoothDevices({

success(res) {

console.log(res)

if (res.devices[0]) {

console.log(ab2hex(res.devices[0].advertisData))

}

}

})

**注意**

* 该接口获取到的设备列表为**蓝牙模块生效期间所有搜索到的蓝牙设备**，若在蓝牙模块使用流程结束后未及时调用 [uni.closeBluetoothAdapter](https://uniapp.dcloud.io/api/system/bluetooth?id=closebluetoothadapter) 释放资源，会存在调用该接口会返回之前的蓝牙使用流程中搜索到的蓝牙设备，可能设备已经不在用户身边，无法连接。
* 蓝牙设备在被搜索到时，系统返回的 name 字段一般为广播包中的 LocalName 字段中的设备名称，而如果与蓝牙设备建立连接，系统返回的 name 字段会改为从蓝牙设备上获取到的 GattName。若需要动态改变设备名称并展示，建议使用 localName 字段。

### [uni.getBluetoothAdapterState(OBJECT)](https://uniapp.dcloud.io/api/system/bluetooth?id=getbluetoothadapterstate)

获取本机蓝牙适配器状态。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| discovering | boolean | 是否正在搜索设备 |
| available | boolean | 蓝牙适配器是否可用 |

#### [错误](https://uniapp.dcloud.io/api/system/bluetooth?id=%e9%94%99%e8%af%af-5)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

**示例代码**

uni.getBluetoothAdapterState({

success(res) {

console.log(res)

}

})

### [uni.closeBluetoothAdapter(OBJECT)](https://uniapp.dcloud.io/api/system/bluetooth?id=closebluetoothadapter)

关闭蓝牙模块。调用该方法将断开所有已建立的连接并释放系统资源。建议在使用蓝牙流程后，与 [uni.openBluetoothAdapter](https://uniapp.dcloud.io/api/system/bluetooth?id=openbluetoothadapter) 成对调用。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/bluetooth?id=%e9%94%99%e8%af%af-6)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

**示例代码**

uni.closeBluetoothAdapter({

success(res) {

console.log(res)

}

})

## 低功耗蓝牙

**低功耗蓝牙 API 平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | x | x | x |

### [uni.writeBLECharacteristicValue(OBJECT)](https://uniapp.dcloud.io/api/system/ble?id=writeblecharacteristicvalue)

向低功耗蓝牙设备特征值中写入二进制数据。注意：必须设备的特征值支持 write 才可以成功调用。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| deviceId | string |  | 是 | 蓝牙设备 id |
| serviceId | string |  | 是 | 蓝牙特征值对应服务的 uuid |
| characteristicId | string |  | 是 | 蓝牙特征值的 uuid |
| value | ArrayBuffer |  | 是 | 蓝牙设备特征值对应的二进制值 |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/ble?id=%e9%94%99%e8%af%af)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

#### [注意](https://uniapp.dcloud.io/api/system/ble?id=%e6%b3%a8%e6%84%8f)

* 并行调用多次会存在写失败的可能性。
* APP不会对写入数据包大小做限制，但系统与蓝牙设备会限制蓝牙4.0单次传输的数据大小，超过最大字节数后会发生写入错误，建议每次写入不超过20字节。
* 若单次写入数据过长，iOS 上存在系统不会有任何回调的情况（包括错误回调）。
* 安卓平台上，在调用 notifyBLECharacteristicValueChange 成功后立即调用 writeBLECharacteristicValue 接口，在部分机型上会发生 10008 系统错误

#### [示例代码](https://uniapp.dcloud.io/api/system/ble?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81)

// 向蓝牙设备发送一个0x00的16进制数据

const buffer = new ArrayBuffer(1)

const dataView = new DataView(buffer)

dataView.setUint8(0, 0)

uni.writeBLECharacteristicValue({

// 这里的 deviceId 需要在 getBluetoothDevices 或 onBluetoothDeviceFound 接口中获取

deviceId,

// 这里的 serviceId 需要在 getBLEDeviceServices 接口中获取

serviceId,

// 这里的 characteristicId 需要在 getBLEDeviceCharacteristics 接口中获取

characteristicId,

// 这里的value是ArrayBuffer类型

value: buffer,

success(res) {

console.log('writeBLECharacteristicValue success', res.errMsg)

}

})

### [uni.readBLECharacteristicValue(OBJECT)](https://uniapp.dcloud.io/api/system/ble?id=readblecharacteristicvalue)

读取低功耗蓝牙设备的特征值的二进制数据值。注意：必须设备的特征值支持 read 才可以成功调用。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| deviceId | string |  | 是 | 蓝牙设备 id |
| serviceId | string |  | 是 | 蓝牙特征值对应服务的 uuid |
| characteristicId | string |  | 是 | 蓝牙特征值的 uuid |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/ble?id=%e9%94%99%e8%af%af-1)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

#### [注意](https://uniapp.dcloud.io/api/system/ble?id=%e6%b3%a8%e6%84%8f-1)

* 并行调用多次会存在读失败的可能性。
* 接口读取到的信息需要在 onBLECharacteristicValueChange 方法注册的回调中获取。

#### [示例代码](https://uniapp.dcloud.io/api/system/ble?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81-1)

// 必须在这里的回调才能获取

uni.onBLECharacteristicValueChange(function (characteristic) {

console.log('characteristic value comed:', characteristic)

})

uni.readBLECharacteristicValue({

// 这里的 deviceId 需要已经通过 createBLEConnection 与对应设备建立链接

deviceId,

// 这里的 serviceId 需要在 getBLEDeviceServices 接口中获取

serviceId,

// 这里的 characteristicId 需要在 getBLEDeviceCharacteristics 接口中获取

characteristicId,

success(res) {

console.log('readBLECharacteristicValue:', res.errCode)

}

})

### [uni.onBLEConnectionStateChange(CALLBACK)](https://uniapp.dcloud.io/api/system/ble?id=onbleconnectionstatechange)

监听低功耗蓝牙连接状态的改变事件。包括开发者主动连接或断开连接，设备丢失，连接异常断开等等

**CALLBACK 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| deviceId | string | 蓝牙设备ID |
| connected | boolean | 是否处于已连接状态 |

#### [示例代码](https://uniapp.dcloud.io/api/system/ble?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81-2)

uni.onBLEConnectionStateChange(function (res) {

// 该方法回调中可以用于处理连接意外断开等异常情况

console.log(`device ${res.deviceId} state has changed, connected: ${res.connected}`)

})

### [uni.onBLECharacteristicValueChange(CALLBACK)](https://uniapp.dcloud.io/api/system/ble?id=onblecharacteristicvaluechange)

监听低功耗蓝牙设备的特征值变化事件。必须先启用 notifyBLECharacteristicValueChange 接口才能接收到设备推送的 notification。

**CALLBACK 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| deviceId | string | 蓝牙设备 id |
| serviceId | string | 蓝牙特征值对应服务的 uuid |
| characteristicId | string | 蓝牙特征值的 uuid |
| value | ArrayBuffer | 特征值最新的值 |

#### [示例代码](https://uniapp.dcloud.io/api/system/ble?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81-3)

// ArrayBuffer转16进度字符串示例

function ab2hex(buffer) {

const hexArr = Array.prototype.map.call(

new Uint8Array(buffer),

function (bit) {

return ('00' + bit.toString(16)).slice(-2)

}

)

return hexArr.join('')

}

uni.onBLECharacteristicValueChange(function (res) {

console.log(`characteristic ${res.characteristicId} has changed, now is ${res.value}`)

console.log(ab2hex(res.value))

})

### [uni.notifyBLECharacteristicValueChange(OBJECT)](https://uniapp.dcloud.io/api/system/ble?id=notifyblecharacteristicvaluechange)

启用低功耗蓝牙设备特征值变化时的 notify 功能，订阅特征值。注意：必须设备的特征值支持 notify 或者 indicate 才可以成功调用。 另外，必须先启用 notifyBLECharacteristicValueChange 才能监听到设备 characteristicValueChange 事件

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| deviceId | string |  | 是 | 蓝牙设备 id |
| serviceId | string |  | 是 | 蓝牙特征值对应服务的 uuid |
| characteristicId | string |  | 是 | 蓝牙特征值的 uuid |
| state | boolean |  | 是 | 是否启用 notify |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/ble?id=%e9%94%99%e8%af%af-2)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

#### [注意](https://uniapp.dcloud.io/api/system/ble?id=%e6%b3%a8%e6%84%8f-2)

* 订阅操作成功后需要设备主动更新特征值的 value，才会触发 [uni.onBLECharacteristicValueChange](https://uniapp.dcloud.io/api/system/ble?id=onblecharacteristicvaluechange) 回调。
* 安卓平台上，在调用 notifyBLECharacteristicValueChange 成功后立即调用 writeBLECharacteristicValue 接口，在部分机型上会发生 10008 系统错误

#### [示例代码](https://uniapp.dcloud.io/api/system/ble?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81-4)

uni.notifyBLECharacteristicValueChange({

state: true, // 启用 notify 功能

// 这里的 deviceId 需要已经通过 createBLEConnection 与对应设备建立链接

deviceId,

// 这里的 serviceId 需要在 getBLEDeviceServices 接口中获取

serviceId,

// 这里的 characteristicId 需要在 getBLEDeviceCharacteristics 接口中获取

characteristicId,

success(res) {

console.log('notifyBLECharacteristicValueChange success', res.errMsg)

}

})

### [uni.getBLEDeviceServices(OBJECT)](https://uniapp.dcloud.io/api/system/ble?id=getbledeviceservices)

获取蓝牙设备所有服务(service)。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| deviceId | string |  | 是 | 蓝牙设备 id |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| services | Array<Object> | 设备服务列表 |

**res.services 的结构**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| uuid | string | 蓝牙设备服务的 uuid |
| isPrimary | boolean | 该服务是否为主服务 |

#### [错误](https://uniapp.dcloud.io/api/system/ble?id=%e9%94%99%e8%af%af-3)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

#### [示例代码](https://uniapp.dcloud.io/api/system/ble?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81-5)

uni.getBLEDeviceServices({

// 这里的 deviceId 需要已经通过 createBLEConnection 与对应设备建立链接

deviceId,

success(res) {

console.log('device services:', res.services)

}

})

### [uni.getBLEDeviceCharacteristics(OBJECT)](https://uniapp.dcloud.io/api/system/ble?id=getbledevicecharacteristics)

获取蓝牙设备某个服务中所有特征值(characteristic)。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| deviceId | string |  | 是 | 蓝牙设备 id |
| serviceId | string |  | 是 | 蓝牙服务 uuid，需要使用 getBLEDeviceServices 获取 |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| characteristics | Array<Object> | 设备服务列表 |

**res.characteristics 的结构**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| uuid | string | 蓝牙设备特征值的 uuid |
| properties | Object | 该特征值支持的操作类型 |

**properties 的结构**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| read | boolean | 该特征值是否支持 read 操作 |
| write | boolean | 该特征值是否支持 write 操作 |
| notify | boolean | 该特征值是否支持 notify 操作 |
| indicate | boolean | 该特征值是否支持 indicate 操作 |

#### [错误](https://uniapp.dcloud.io/api/system/ble?id=%e9%94%99%e8%af%af-4)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

#### [示例代码](https://uniapp.dcloud.io/api/system/ble?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81-6)

uni.getBLEDeviceCharacteristics({

// 这里的 deviceId 需要已经通过 createBLEConnection 与对应设备建立链接

deviceId,

// 这里的 serviceId 需要在 getBLEDeviceServices 接口中获取

serviceId,

success(res) {

console.log('device getBLEDeviceCharacteristics:', res.characteristics)

}

})

### [uni.createBLEConnection(OBJECT)](https://uniapp.dcloud.io/api/system/ble?id=createbleconnection)

连接低功耗蓝牙设备。

若APP在之前已有搜索过某个蓝牙设备，并成功建立连接，可直接传入之前搜索获取的 deviceId 直接尝试连接该设备，无需进行搜索操作。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| deviceId | string |  | 是 | 用于区分设备的 id |
| timeout | number |  | 否 | 超时时间，单位ms，不填表示不会超时 |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/ble?id=%e9%94%99%e8%af%af-5)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

#### [注意](https://uniapp.dcloud.io/api/system/ble?id=%e6%b3%a8%e6%84%8f-3)

* 请保证尽量成对的调用 createBLEConnection 和 closeBLEConnection 接口。安卓如果多次调用 createBLEConnection 创建连接，有可能导致系统持有同一设备多个连接的实例，导致调用 closeBLEConnection 的时候并不能真正的断开与设备的连接。
* 蓝牙连接随时可能断开，建议监听 [uni.onBLEConnectionStateChange](https://uniapp.dcloud.io/api/system/ble?id=onbleconnectionstatechange) 回调事件，当蓝牙设备断开时按需执行重连操作
* 若对未连接的设备或已断开连接的设备调用数据读写操作的接口，会返回 10006 错误，建议进行重连操作。

#### [示例代码](https://uniapp.dcloud.io/api/system/ble?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81-7)

uni.createBLEConnection({

// 这里的 deviceId 需要已经通过 createBLEConnection 与对应设备建立链接

deviceId,

success(res) {

console.log(res)

}

})

### [uni.closeBLEConnection(OBJECT)](https://uniapp.dcloud.io/api/system/ble?id=closebleconnection)

断开与低功耗蓝牙设备的连接。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| deviceId | string |  | 是 | 用于区分设备的 id |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/ble?id=%e9%94%99%e8%af%af-6)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 10000 | not init | 未初始化蓝牙适配器 |
| 10001 | not available | 当前蓝牙适配器不可用 |
| 10002 | no device | 没有找到指定设备 |
| 10003 | connection fail | 连接失败 |
| 10004 | no service | 没有找到指定服务 |
| 10005 | no characteristic | 没有找到指定特征值 |
| 10006 | no connection | 当前连接已断开 |
| 10007 | property not support | 当前特征值不支持此操作 |
| 10008 | system error | 其余所有系统上报的异常 |
| 10009 | system not support | Android 系统特有，系统版本低于 4.3 不支持 BLE |

#### [示例代码](https://uniapp.dcloud.io/api/system/ble?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81-8)

uni.closeBLEConnection({

deviceId,

success(res) {

console.log(res)

}

})

## iBeacon

**iBeacon API 平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | x | x | x |

### [uni.onBeaconServiceChange(CALLBACK)](https://uniapp.dcloud.io/api/system/ibeacon?id=onbeaconservicechange)

监听 iBeacon 服务状态变化事件

**CALLBACK 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| available | boolean | 服务目前是否可用 |
| discovering | boolean | 目前是否处于搜索状态 |

### [uni.onBeaconUpdate(CALLBACK)](https://uniapp.dcloud.io/api/system/ibeacon?id=onbeaconupdate)

监听 iBeacon 设备更新事件

**CALLBACK 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| beacons | Array<[IBeaconInfo](https://uniapp.dcloud.io/api/system/ibeacon?id=ibeaconinfo)> | 当前搜寻到的所有 iBeacon 设备列表 |

### [uni.getBeacons(OBJECT)](https://uniapp.dcloud.io/api/system/ibeacon?id=getbeacons)

获取所有已搜索到的 iBeacon 设备

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| beacons | Array<[IBeaconInfo](https://uniapp.dcloud.io/api/system/ibeacon?id=ibeaconinfo)> | iBeacon 设备列表 |

#### [错误](https://uniapp.dcloud.io/api/system/ibeacon?id=%e9%94%99%e8%af%af)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 11000 | unsupport | 系统或设备不支持 |
| 11001 | bluetooth service unavailable | 蓝牙服务不可用 |
| 11002 | location service unavailable | 位置服务不可用 |
| 11003 | already start | 已经开始搜索 |

### [uni.startBeaconDiscovery(OBJECT)](https://uniapp.dcloud.io/api/system/ibeacon?id=startbeacondiscovery)

开始搜索附近的 iBeacon 设备

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| uuids | Array＜String＞ |  | 是 | iBeacon 设备广播的 uuid 列表 |
| ignoreBluetoothAvailable | boolean | false | 否 | 是否校验蓝牙开关，仅在 iOS 下有效 |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/ibeacon?id=%e9%94%99%e8%af%af-1)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 11000 | unsupport | 系统或设备不支持 |
| 11001 | bluetooth service unavailable | 蓝牙服务不可用 |
| 11002 | location service unavailable | 位置服务不可用 |
| 11003 | already start | 已经开始搜索 |

#### [示例代码](https://uniapp.dcloud.io/api/system/ibeacon?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81)

uni.startBeaconDiscovery({

success(res) { }

})

### [uni.stopBeaconDiscovery(OBJECT)](https://uniapp.dcloud.io/api/system/ibeacon?id=stopbeacondiscovery)

停止搜索附近的 iBeacon 设备

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

#### [错误](https://uniapp.dcloud.io/api/system/ibeacon?id=%e9%94%99%e8%af%af-2)

| **错误码** | **错误信息** | **说明** |
| --- | --- | --- |
| 0 | ok | 正常 |
| 11000 | unsupport | 系统或设备不支持 |
| 11001 | bluetooth service unavailable | 蓝牙服务不可用 |
| 11002 | location service unavailable | 位置服务不可用 |
| 11003 | already start | 已经开始搜索 |

### [IBeaconInfo](https://uniapp.dcloud.io/api/system/ibeacon?id=ibeaconinfo)

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| uuid | string | iBeacon 设备广播的 uuid |
| major | string | iBeacon 设备的主 id |
| minor | string | iBeacon 设备的次 id |
| proximity | number | 表示设备距离的枚举值 |
| accuracy | number | iBeacon 设备的距离 |
| rssi | number | 表示设备的信号强度 |

## Wi-Fi

#### [Wi-Fi](https://uniapp.dcloud.io/api/system/wifi?id=wi-fi)

仅微信小程序平台、App平台、头条小程序支持，各平台开发方式暂未统一，使用时需注意用[条件编译](https://uniapp.dcloud.io/platform)调用不同平台的代码。

微信小程序平台实现参考：[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/wx.startWifi.html)

头条小程序的wifi API参考：[规范详情](https://developer.toutiao.com/dev/cn/mini-app/develop/api/device/wi-fi/getconnectedwifi)

App 平台实现参考

**安卓：**

* [获取WIFI列表](https://ask.dcloud.net.cn/question/12113)

**ios：**

* [打开ios的WIFI设置页面](https://ask.dcloud.net.cn/question/7797)

## 电量

#### [电量](https://uniapp.dcloud.io/api/system/batteryInfo?id=%e7%94%b5%e9%87%8f)

电量API暂未统一，需分平台条件编译编写。

* 微信小程序平台：[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/wx.getBatteryInfoSync.html)
* 百度小程序平台：[规范详情](https://smartprogram.baidu.com/docs/develop/api/device_battery/#swan-getBatteryInfo/)
* 支付宝小程序平台：[规范详情](https://docs.alipay.com/mini/api/nrnziy)
* QQ小程序平台：[规范详情](https://q.qq.com/wiki/develop/miniprogram/API/equipment/ibeacon_battery.html)
* App平台：使用 Native.js，[参考](https://ask.dcloud.net.cn/article/992" \t "_blank)
* H5平台：使用 navigator.battery API

## NFC

#### [NFC](https://uniapp.dcloud.io/api/system/nfc?id=nfc)

仅微信小程序平台、App平台支持，各平台开发方式暂未统一，使用时需注意用[条件编译](https://uniapp.dcloud.io/platform)调用不同平台的代码。

* 微信小程序平台实现参考：[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/wx.startHCE.html)
* App 平台通过Native.js实现，**安卓：**[NFC数据读取](https://ask.dcloud.net.cn/question/6726)

## 设备方向

#### [监听设备方向变化](https://uniapp.dcloud.io/api/system/deviceMotion?id=%e7%9b%91%e5%90%ac%e8%ae%be%e5%a4%87%e6%96%b9%e5%90%91%e5%8f%98%e5%8c%96)

仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/device/motion/wx.startDeviceMotionListening.html)

在App平台，也可以通过监听窗体大小变化onResize来实现此类需求。[详见](https://uniapp.dcloud.io/collocation/frame/lifecycle?id=%e9%a1%b5%e9%9d%a2%e7%94%9f%e5%91%bd%e5%91%a8%e6%9c%9f)

## 生物认证

### [生物认证说明](https://uniapp.dcloud.io/api/system/authentication?id=%e7%94%9f%e7%89%a9%e8%ae%a4%e8%af%81%e8%af%b4%e6%98%8e)

生物认证，又称活体检测。它包含指纹识别、人脸识别这两部分。即通过人体身体特征来进行身份认证识别。

### [uni.startSoterAuthentication(OBJECT)](https://uniapp.dcloud.io/api/system/authentication?id=startsoterauthentication)

开始 SOTER 生物认证。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √（2.3.8+） | x | √ | x | x | x | x |

App端在2.3.8版以前，可在插件市场获取[指纹相关插件](https://ext.dcloud.net.cn/plugin?id=358)。

**OBJECT参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| requestAuthModes | Array |  | 是 | 请求使用的可接受的生物认证方式 | APP、微信小程序 |
| challenge | String |  | 是 | 挑战因子。挑战因子为调用者为此次生物鉴权准备的用于签名的字符串关键识别信息，将作为 resultJSON 的一部分，供调用者识别本次请求。例如：如果场景为请求用户对某订单进行授权确认，则可以将订单号填入此参数。 | 微信小程序 |
| authContent | String | '' | 否 | 验证描述，即识别过程中显示在界面上的对话框提示内容 | APP、微信小程序 |
| success | Function |  | 否 | 接口调用成功的回调函数 |  |
| fail | Function |  | 否 | 接口调用失败的回调函数 |  |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**OBJECT.requestAuthModes说明**

| **值** | **说明** |
| --- | --- |
| fingerPrint | 指纹识别 |
| facial | 人脸识别 |

注意：

* App端指纹识别，Android平台从Android6.0起才提供了官方API，uni-app也是从Android6起支持。对于更低版本的安卓，某些rom私有的指纹识别API，uni-app并不支持。
* App端人脸识别，iOS平台使用自带的faceID，而Android平台需要依赖三方SDK方可实现，可在插件市场搜索[人脸识别](https://ext.dcloud.net.cn/search?q=%E4%BA%BA%E8%84%B8%E8%AF%86%E5%88%AB)插件

**OBJECT.success返回值说明**

| **属性** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| authMode | string | 生物认证方式 | APP、微信小程序 |
| resultJSON | string | 在设备安全区域（TEE）内获得的本机安全信息（如TEE名称版本号等以及防重放参数）以及本次认证信息（仅Android支持，本次认证的指纹ID）。具体说明见下文 | 微信小程序 |
| resultJSONSignature | string | 用SOTER安全密钥对 resultJSON 的签名(SHA256 with RSA/PSS, saltlen=20) | 微信小程序 |
| errCode | number | 错误码 |  |
| errMsg | string | 错误信息 |  |

**resultJSON说明**

此数据为设备TEE中，将传入的challenge和TEE内其他安全信息组成的数据进行组装而来的JSON，对下述字段的解释如下表。例子如下：

| **字段名** | **说明** |
| --- | --- |
| raw | 调用者传入的challenge |
| fid | （仅Android支持）本次生物识别认证的生物信息编号（如指纹识别则是指纹信息在本设备内部编号） |
| counter | 防重放特征参数 |
| tee\_n | TEE名称（如高通或者trustonic等） |
| tee\_v | TEE版本号 |
| fp\_n | 指纹以及相关逻辑模块提供商（如FPC等） |
| fp\_v | 指纹以及相关模块版本号 |
| cpu\_id | 机器唯一识别ID |
| uid | 概念同Android系统定义uid，即应用程序编号 |

**错误码说明**

| **错误码** | **错误码说明** |
| --- | --- |
| 0 | 识别成功 |
| 90001 | 本设备不支持生物认证 |
| 90002 | 用户未授权使用该生物认证接口 |
| 90003 | 请求使用的生物认证方式不支持 |
| 90004 | 未传入challenge或challenge长度过长（最长512字符） |
| 90005 | auth\_content长度超过限制（最长42个字符） |
| 90007 | 内部错误 |
| 90008 | 用户取消授权 |
| 90009 | 识别失败 |
| 90010 | 重试次数过多被冻结 |
| 90011 | 用户未录入所选识别方式 |

### [uni.checkIsSupportSoterAuthentication(OBJECT)](https://uniapp.dcloud.io/api/system/authentication?id=checkissupportsoterauthentication)

获取本机支持的 SOTER 生物认证方式

**OBJECT参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**OBJECT.success返回值说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| supportMode | Array | 该设备支持的可被SOTER识别的生物识别方式 |

### [uni.checkIsSoterEnrolledInDevice(OBJECT)](https://uniapp.dcloud.io/api/system/authentication?id=checkissoterenrolledindevice)

获取设备内是否录入如指纹等生物信息的接口

**OBJECT参数说明**

|属性 |类型 |默认值 |必填 |说明 | |checkAuthMode|string | |是 |认证方式 | |success |function | |否 |接口调用成功的回调函数 | |fail |function | |否 |接口调用失败的回调函数 | |complete |function | |否 |接口调用结束的回调函数（调用成功、失败都会执行） |

**OBJECT.checkAuthMode合法值**

| **值** | **说明** |
| --- | --- |
| fingerPrint | 指纹识别 |
| facial | 人脸识别 |

**OBJECT.success返回值说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| isEnrolled | boolean | 是否已录入信息 |
| errMsg | string | 错误信息 |

#### [代码示例](https://uniapp.dcloud.io/api/system/authentication?id=%e4%bb%a3%e7%a0%81%e7%a4%ba%e4%be%8b)

<template>

<view class="content">

<button type="primary" @click="checkIsSupportSoterAuthentication">检查支持的认证方式</button>

<button type="primary" @click="checkIsSoterEnrolledInDeviceFingerPrint">检查是否录入指纹</button>

<button type="primary" @click="checkIsSoterEnrolledInDeviceFaceID">检查是否录入FaceID</button>

<button type="primary" @click="startSoterAuthenticationFingerPrint">开始指纹认证</button>

<button type="primary" @click="startSoterAuthenticationFaceID">开始FaceID认证</button>

</view>

</template>

<script>

export default {

data() {

return {

}

},

onLoad() {

},

methods: {

checkIsSupportSoterAuthentication() {

uni.checkIsSupportSoterAuthentication({

success(res) {

console.log(res);

},

fail(err) {

console.log(err);

},

complete(res) {

console.log(res);

}

})

},

checkIsSoterEnrolledInDeviceFingerPrint() {

uni.checkIsSoterEnrolledInDevice({

checkAuthMode: 'fingerPrint',

success(res) {

console.log(res);

},

fail(err) {

console.log(err);

},

complete(res) {

console.log(res);

}

})

},

checkIsSoterEnrolledInDeviceFaceID() {

uni.checkIsSoterEnrolledInDevice({

checkAuthMode: 'facial',

success(res) {

console.log(res);

},

fail(err) {

console.log(err);

},

complete(res) {

console.log(res);

}

})

},

startSoterAuthenticationFingerPrint() {

uni.startSoterAuthentication({

requestAuthModes: ['fingerPrint'],

challenge: '123456',

authContent: '请用指纹解锁',

success(res) {

console.log(res);

},

fail(err) {

console.log(err);

},

complete(res) {

console.log(res);

}

})

},

startSoterAuthenticationFaceID() {

uni.startSoterAuthentication({

requestAuthModes: ['facial'],

challenge: '123456',

authContent: '请用FaceID解锁',

success(res) {

console.log(res);

},

fail(err) {

console.log(err);

},

complete(res) {

console.log(res);

}

})

}

}

}

</script>

<style>

button {

width: 200px;

margin: 20px auto;

}

</style>

#### [注意事项](https://uniapp.dcloud.io/api/system/authentication?id=%e6%b3%a8%e6%84%8f%e4%ba%8b%e9%a1%b9)

* App端自2.3.8版本起开始支持生物认证，更低版本或想使用指纹功能，可在插件市场获取[插件](https://ext.dcloud.net.cn/plugin?id=358)
* App端的人脸识别，仅支持iOS端的faceID。Android端需要依赖三方SDK方可实现，可在插件市场搜索[人脸识别](https://ext.dcloud.net.cn/search?q=%E4%BA%BA%E8%84%B8%E8%AF%86%E5%88%AB)插件
* App端打包时，注意需要在manifest的模块中选择指纹和faceID，否则打包后无法运行相关功能。
* hello uni-app已经集成相关示例，最新版HBuilderX新建新版hello uni-app示例项目真机运行可见，在API-设备-生物认证里。
* 微信小程序如果使用腾讯云的SDK，可参考[网友分享](https://segmentfault.com/a/1190000020102601)
* 支付宝小程序只支持人脸识别，[规范详情](https://docs.alipay.com/mini/api/facecapture)
* 百度小程序只支持人脸识别，[规范详情](https://smartprogram.baidu.com/docs/develop/api/ai_face/#swan-ai-faceDetect/)

# Worke

#### [Worker](https://uniapp.dcloud.io/api/worker?id=worker)

目前需分平台编写

* 微信小程序：[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/worker/wx.createWorker.html)
* 头条小程序：[规范详情](https://developer.toutiao.com/docs/game/worker/tt.createWorker.html)
* QQ小程序：[规范详情](https://q.qq.com/wiki/develop/miniprogram/API/worker/worker.html)
* H5：标准H5的worker仍然可以使用

# 键盘

#### [uni.hideKeyboard()](https://uniapp.dcloud.io/api/key?id=hidekeyboard)

隐藏软键盘

隐藏已经显示的软键盘，如果软键盘没有显示则不做任何操作。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | x | √ | √ |

#### [uni.onKeyboardHeightChange(CALLBACK)](https://uniapp.dcloud.io/api/key?id=onkeyboardheightchange)

监听键盘高度变化

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| HBuilderX 2.2.3+ | x | 基础库2.7+ | x | x | x | √ |

**CALLBACK 返回参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| height | Number | 键盘高度 |

**示例代码**

uni.onKeyboardHeightChange(res => {

console.log(res.height)

})

# 界面

## 交互反馈

### [uni.showToast(OBJECT)](https://uniapp.dcloud.io/api/ui/prompt?id=showtoast)

显示消息提示框。

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| title | String | 是 | 提示的内容，长度与 icon 取值有关。 |  |
| icon | String | 否 | 图标，有效值详见下方说明。 |  |
| image | String | 否 | 自定义图标的本地路径 | App、H5、微信小程序、百度小程序 |
| mask | Boolean | 否 | 是否显示透明蒙层，防止触摸穿透，默认：false | App、微信小程序 |
| duration | Number | 否 | 提示的延迟时间，单位毫秒，默认：1500 |  |
| position | String | 否 | 纯文本轻提示显示位置，填写有效值后只有 title 属性生效， 有效值详见下方说明。 | App |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**icon 值说明**

| **值** | **说明** | **平台差异说明** |
| --- | --- | --- |
| success | 显示成功图标，此时 title 文本最多显示 7 个汉字长度。默认值 |  |
| loading | 显示加载图标，此时 title 文本最多显示 7 个汉字长度。 | 支付宝小程序不支持 |
| none | 不显示图标，此时 title 文本在小程序最多可显示两行，App仅支持单行显示。 |  |

**示例**

uni.showToast({

title: '标题',

duration: 2000

});

**position 值说明（仅App生效）**

| **值** | **说明** |
| --- | --- |
| top | 居上显示 |
| center | 居中显示 |
| bottom | 居底显示 |

**Tips**

App端可通过[plus.nativeUI.toast API](https://www.html5plus.org/doc/zh_cn/nativeui.html" \l "plus.nativeUI.toast" \t "_blank)实现更多功能。

### [uni.hideToast()](https://uniapp.dcloud.io/api/ui/prompt?id=hidetoast)

隐藏消息提示框。

**示例**

uni.hideToast();

### [uni.showLoading(OBJECT)](https://uniapp.dcloud.io/api/ui/prompt?id=showloading)

显示 loading 提示框, 需主动调用 [uni.hideLoading](https://uniapp.dcloud.io/api/ui/prompt?id=hideloading) 才能关闭提示框。

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| title | String | 是 | 提示的内容 |  |
| mask | Boolean | 否 | 是否显示透明蒙层，防止触摸穿透，默认：false | App、微信小程序、百度小程序 |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**示例**

uni.showLoading({

title: '加载中'

});

### [uni.hideLoading()](https://uniapp.dcloud.io/api/ui/prompt?id=hideloading)

隐藏 loading 提示框。

**示例**

uni.showLoading({

title: '加载中'

});

setTimeout(function () {

uni.hideLoading();

}, 2000);

### [uni.showModal(OBJECT)](https://uniapp.dcloud.io/api/ui/prompt?id=showmodal)

显示模态弹窗，类似于标准 html 的消息框：alert、confirm。

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| title | String | 否 | 提示的标题 |  |
| content | String | 否 | 提示的内容 |  |
| showCancel | Boolean | 否 | 是否显示取消按钮，默认为 true |  |
| cancelText | String | 否 | 取消按钮的文字，默认为"取消"，最多 4 个字符 |  |
| cancelColor | HexColor | 否 | 取消按钮的文字颜色，默认为"#000000" | H5、微信小程序、百度小程序 |
| confirmText | String | 否 | 确定按钮的文字，默认为"确定"，最多 4 个字符 |  |
| confirmColor | HexColor | 否 | 确定按钮的文字颜色，H5平台默认为"#007aff"，微信小程序平台默认为"#3CC51F"，百度小程序平台默认为"#3c76ff" | H5、微信小程序、百度小程序 |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**注意**

* 钉钉小程序真机与模拟器表现有差异，真机title，content均为必填项

**success返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| confirm | Boolean | 为 true 时，表示用户点击了确定按钮 |
| cancel | Boolean | 为 true 时，表示用户点击了取消（用于 Android 系统区分点击蒙层关闭还是点击取消按钮关闭） |

**Tips：**

* 在 App 下可以使用 [原生增强提示框插件](https://ext.dcloud.net.cn/plugin?id=36) 来解决 App 无法设置 cancelColor、confirmColor的问题。

**示例**

uni.showModal({

title: '提示',

content: '这是一个模态弹窗',

success: function (res) {

if (res.confirm) {

console.log('用户点击确定');

} else if (res.cancel) {

console.log('用户点击取消');

}

}

});

### [uni.showActionSheet(OBJECT)](https://uniapp.dcloud.io/api/ui/prompt?id=showactionsheet)

​显示操作菜单

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| itemList | Array<String> | 是 | 按钮的文字数组 | 微信、百度、头条小程序数组长度最大为6个 |
| itemColor | HexColor | 否 | 按钮的文字颜色，字符串格式，默认为"#000000" | 头条小程序不支持 |
| success | Function | 否 | 接口调用成功的回调函数，详见返回参数说明 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**success返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| tapIndex | Number | 用户点击的按钮，从上到下的顺序，从0开始 |

**示例**

uni.showActionSheet({

itemList: ['A', 'B', 'C'],

success: function (res) {

console.log('选中了第' + (res.tapIndex + 1) + '个按钮');

},

fail: function (res) {

console.log(res.errMsg);

}

});

**Tips**

App端实现原生的、复杂的底部图文菜单，例如分享菜单，可参考<https://ext.dcloud.net.cn/plugin?id=69>

**注意**

* 本章的所有弹出控件都是原生控件，层级最高，可覆盖video、map、tabbar等原生控件。
* [uni-app插件市场](https://ext.dcloud.net.cn/)有很多封装好的前端组件，但注意前端组件层级不是最高，无法覆盖原生组件，除非使用cover-view或nvue。
* App端还有原生的[prompt API](https://www.html5plus.org/doc/zh_cn/nativeui.html#plus.nativeUI.prompt)

## 设置导航条

### [uni.setNavigationBarTitle(OBJECT)](https://uniapp.dcloud.io/api/ui/navigationbar?id=setnavigationbartitle)

动态设置当前页面的标题。

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| title | String | 是 | 页面标题 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.setNavigationBarTitle({

title: '新的标题'

});

### [uni.setNavigationBarColor(OBJECT)](https://uniapp.dcloud.io/api/ui/navigationbar?id=setnavigationbarcolor)

设置页面导航条颜色。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | √ | √ |

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| frontColor | String | 是 | 前景颜色值，包括按钮、标题、状态栏的颜色，仅支持 #ffffff 和 #000000 | App、H5、微信小程序、百度小程序 |
| backgroundColor | String | 是 | 背景颜色值，有效值为十六进制颜色 |  |
| animation | Object | 否 | 动画效果，{duration,timingFunc} | 微信小程序、百度小程序 |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**注意**

* Android 上的 backgroundColor 参数有限制，黑色大于 rgb(30,30,30), 白色小于 rgb(235,235,235)

**animation 结构**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| duration | number | 0 | 否 | 动画变化时间，单位 ms |
| timingFunc | String | 'linear' | 否 | 动画变化方式 |

**animation.timingFunc 有效值**

| **值** | **说明** |
| --- | --- |
| linear | 动画从头到尾的速度是相同的。 |
| easeIn | 动画以低速开始 |
| easeOut | 动画以低速结束。 |
| easeInOut | 动画以低速开始和结束。 |

**success返回参数说明**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 调用结果 |

**示例**

uni.setNavigationBarColor({

frontColor: '#ffffff',

backgroundColor: '#ff0000',

animation: {

duration: 400,

timingFunc: 'easeIn'

}

})

### [uni.showNavigationBarLoading(OBJECT)](https://uniapp.dcloud.io/api/ui/navigationbar?id=shownavigationbarloading)

在当前页面显示导航条加载动画。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | √ | √ | √ | √ | x | √ |

App平台调用此API时会在屏幕中间悬浮显示loading

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.showNavigationBarLoading()

### [uni.hideNavigationBarLoading(OBJECT)](https://uniapp.dcloud.io/api/ui/navigationbar?id=hidenavigationbarloading)

在当前页面隐藏导航条加载动画。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | √ | √ | √ | √ | x | √ |

App平台调用此API时会关闭屏幕中间悬浮显示的loading

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.hideNavigationBarLoading()

### [uni.hideHomeButton(OBJECT)](https://uniapp.dcloud.io/api/ui/navigationbar?id=hidehomebutton)

隐藏返回首页按钮。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | x | x | x |

**OBJECT参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例**

uni.hideHomeButton()

**说明**

* 微信小程序自基础库版本2.8.3开始支持
* 当用户打开的小程序最底层页面是非首页时，默认展示“返回首页”按钮，开发者可在页面onShow中调用hideHomeButton进行隐藏。

## 设置TabBar

### [uni.setTabBarItem(OBJECT)](https://uniapp.dcloud.io/api/ui/tabbar?id=settabbaritem)

动态设置 tabBar 某一项的内容

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**OBJECT参数说明：**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| index | number |  | 是 | tabBar 的哪一项，从左边算起 |
| text | String |  | 否 | tab 上的按钮文字 |
| iconPath | String |  | 否 | 图片路径，icon 大小限制为 40kb，建议尺寸为 81px \* 81px，当 postion 为 top 时，此参数无效，不支持网络图片 |
| selectedIconPath | String |  | 否 | 选中时的图片路径，icon 大小限制为 40kb，建议尺寸为 81px \* 81px ，当 postion 为 top 时，此参数无效 |
| success | Funtion |  | 否 | 接口调用成功的回调函数 |
| fail | Funtion |  | 否 | 接口调用失败的回调函数 |
| complete | Funtion |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例代码**

uni.setTabBarItem({

index: 0,

text: 'text',

iconPath: '/path/to/iconPath',

selectedIconPath: '/path/to/selectedIconPath'

})

### [uni.setTabBarStyle(OBJECT)](https://uniapp.dcloud.io/api/ui/tabbar?id=settabbarstyle)

动态设置 tabBar 的整体样式

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**OBJECT参数说明：**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| color | String |  | 是 | tab 上的文字默认颜色，HexColor |
| selectedColor | String |  | 是 | tab 上的文字选中时的颜色，HexColor |
| backgroundColor | String |  | 是 | tab 的背景色，HexColor |
| borderStyle | String |  | 是 | tabBar上边框的颜色， 仅支持 black/white |
| success | Funtion |  | 否 | 接口调用成功的回调函数 |
| fail | Funtion |  | 否 | 接口调用失败的回调函数 |
| complete | Funtion |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例代码**

uni.setTabBarStyle({

color: '#FF0000',

selectedColor: '#00FF00',

backgroundColor: '#0000FF',

borderStyle: 'white'

})

### [uni.hideTabBar(OBJECT)](https://uniapp.dcloud.io/api/ui/tabbar?id=hidetabbar)

隐藏 tabBar

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**OBJECT参数说明：**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| animation | boolean | false | 否 | 是否需要动画效果，仅微信小程序和百度小程序支持 |
| success | Funtion |  | 否 | 接口调用成功的回调函数 |
| fail | Funtion |  | 否 | 接口调用失败的回调函数 |
| complete | Funtion |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [uni.showTabBar(OBJECT)](https://uniapp.dcloud.io/api/ui/tabbar?id=showtabbar)

显示 tabBar

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**OBJECT参数说明：**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| animation | boolean | false | 否 | 是否需要动画效果，仅微信小程序和百度小程序支持 |
| success | Funtion |  | 否 | 接口调用成功的回调函数 |
| fail | Funtion |  | 否 | 接口调用失败的回调函数 |
| complete | Funtion |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [uni.setTabBarBadge(OBJECT)](https://uniapp.dcloud.io/api/ui/tabbar?id=settabbarbadge)

为 tabBar 某一项的右上角添加文本。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**OBJECT参数说明：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| index | Number | 是 | tabBar的哪一项，从左边算起 |
| text | String | 是 | 显示的文本，不超过 3 个半角字符 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例代码**

uni.setTabBarBadge({

index: 0,

text: '1'

})

### [uni.removeTabBarBadge(OBJECT)](https://uniapp.dcloud.io/api/ui/tabbar?id=removetabbarbadge)

移除 tabBar 某一项右上角的文本。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**OBJECT参数说明：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| index | Number | 是 | tabBar的哪一项，从左边算起 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [uni.showTabBarRedDot(OBJECT)](https://uniapp.dcloud.io/api/ui/tabbar?id=showtabbarreddot)

显示 tabBar 某一项的右上角的红点。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**OBJECT参数说明：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| index | Number | 是 | tabBar的哪一项，从左边算起 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [uni.hideTabBarRedDot(OBJECT)](https://uniapp.dcloud.io/api/ui/tabbar?id=hidetabbarreddot)

隐藏 tabBar 某一项的右上角的红点。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | x | √ |

**OBJECT参数说明：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| index | Number | 是 | tabBar的哪一项，从左边算起 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [uni.onTabBarMidButtonTap(CALLBACK)](https://uniapp.dcloud.io/api/ui/tabbar?id=ontabbarmidbuttontap)

监听中间按钮的点击事件

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √（HBuilderX 2.3.4+） | x | x | x | x | x | x |

**Tip**

* tabbar是原生的，层级高于前端元素
* [uni-app插件市场](https://ext.dcloud.net.cn/search?q=%E5%BA%95%E9%83%A8%E5%9B%BE%E6%A0%87%E8%8F%9C%E5%8D%95)有封装的前端tabbar，但性能不如原生tabbar
* 如果想要一个中间带+号的tabbar，在HBuilderX中新建uni-app项目、选择 底部选项卡 模板

## 背景

### [uni.setBackgroundColor(OBJECT)](https://uniapp.dcloud.io/api/ui/bgcolor?id=setbackgroundcolor)

动态设置窗口的背景色。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | √ |

**参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| backgroundColor | String |  | 否 | 窗口的背景色，必须为十六进制颜色值 |
| backgroundColorTop | String |  | 否 | 顶部窗口的背景色，必须为十六进制颜色值，仅 iOS 支持 |
| backgroundColorBottom | String |  | 否 | 底部窗口的背景色，必须为十六进制颜色值，仅 iOS 支持 |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**代码示例**

uni.setBackgroundColor({

backgroundColor: '#ffffff',

backgroundColorTop: '#222222',

backgroundColorBottom: '#333333'

});

### [uni.setBackgroundTextStyle(OBJECT)](https://uniapp.dcloud.io/api/ui/bgcolor?id=setbackgroundtextstyle)

动态设置下拉背景字体、loading 图的样式。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | √ |

**参数说明**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| textStyle | String | 是 | 下拉背景字体、loading 图的样式，值为：dark、light |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**代码示例**

uni.setBackgroundTextStyle({

textStyle: 'dark' // 下拉背景字体、loading 图的样式为dark

})

## 动画

### [uni.createAnimation(OBJECT)](https://uniapp.dcloud.io/api/ui/animation?id=createanimation)

创建一个动画实例 [animation](https://uniapp.dcloud.io/api/ui/animation?id=animation)。调用实例的方法来描述动画。最后通过动画实例的export方法导出动画数据传递给组件的animation属性。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | HBuilderX 2.0.4+ | √ | √ | √ | √ | √ |

**注意:**

* export 方法每次调用后会清掉之前的动画操作
* nvue 暂不支持

**OBJECT参数说明：**

| **参数** | **类型** | **必填** | **默认值** | **说明** |
| --- | --- | --- | --- | --- |
| duration | Integer | 否 | 400 | 动画持续时间，单位ms |
| timingFunction | String | 否 | "linear" | 定义动画的效果 |
| delay | Integer | 否 | 0 | 动画延迟时间，单位 ms |
| transformOrigin | String | 否 | "50% 50% 0" | 设置transform-origin |

**timingFunction 有效值：**

| **值** | **说明** |
| --- | --- |
| linear | 动画从头到尾的速度是相同的 |
| ease | 动画以低速开始，然后加快，在结束前变慢 |
| ease-in | 动画以低速开始 |
| ease-in-out | 动画以低速开始和结束 |
| ease-out | 动画以低速结束 |
| step-start | 动画第一帧就跳至结束状态直到结束 |
| step-end | 动画一直保持开始状态，最后一帧跳到结束状态 |

var animation = uni.createAnimation({

transformOrigin: "50% 50%",

duration: 1000,

timingFunction: "ease",

delay: 0

})

**animation**

动画实例可以调用以下方法来描述动画，调用结束后会返回自身，支持链式调用的写法。

**animation 对象的方法列表：**

样式：

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| opacity | value | 透明度，参数范围 0~1 |
| backgroundColor | color | 颜色值 |
| width | length | 长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值 |
| height | length | 长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值 |
| top | length | 长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值 |
| left | length | 长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值 |
| bottom | length | 长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值 |
| right | length | 长度值，如果传入 Number 则默认使用 px，可传入其他自定义单位的长度值 |

旋转：

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| rotate | deg | deg的范围-180~180，从原点顺时针旋转一个deg角度 |
| rotateX | deg | deg的范围-180~180，在X轴旋转一个deg角度 |
| rotateY | deg | deg的范围-180~180，在Y轴旋转一个deg角度 |
| rotateZ | deg | deg的范围-180~180，在Z轴旋转一个deg角度 |
| rotate3d | (x,y,z,deg) | 同[transform-function rotate3d](https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/rotate3d" \t "_blank) |

缩放：

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| scale | sx,[sy] | 一个参数时，表示在X轴、Y轴同时缩放sx倍数；两个参数时表示在X轴缩放sx倍数，在Y轴缩放sy倍数 |
| scaleX | sx | 在X轴缩放sx倍数 |
| scaleY | sy | 在Y轴缩放sy倍数 |
| scaleZ | sz | 在Z轴缩放sy倍数 |
| scale3d | (sx,sy,sz) | 在X轴缩放sx倍数，在Y轴缩放sy倍数，在Z轴缩放sz倍数 |

偏移：

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| translate | tx,[ty] | 一个参数时，表示在X轴偏移tx，单位px；两个参数时，表示在X轴偏移tx，在Y轴偏移ty，单位px。 |
| translateX | tx | 在X轴偏移tx，单位px |
| translateY | ty | 在Y轴偏移tx，单位px |
| translateZ | tz | 在Z轴偏移tx，单位px |
| translate3d | (tx,ty,tz) | 在X轴偏移tx，在Y轴偏移ty，在Z轴偏移tz，单位px |

倾斜：

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| skew | ax,[ay] | 参数范围-180~180；一个参数时，Y轴坐标不变，X轴坐标延顺时针倾斜ax度；两个参数时，分别在X轴倾斜ax度，在Y轴倾斜ay度 |
| skewX | ax | 参数范围-180~180；Y轴坐标不变，X轴坐标延顺时针倾斜ax度 |
| skewY | ay | 参数范围-180~180；X轴坐标不变，Y轴坐标延顺时针倾斜ay度 |

矩阵变形：

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| matrix | (a,b,c,d,tx,ty) | 同 [transform-function matrix](https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/matrix) |
| matrix3d |  | 同[transform-function matrix3d](https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/matrix3d" \t "_blank) |

**动画队列**

调用动画操作方法后要调用 step() 来表示一组动画完成，可以在一组动画中调用任意多个动画方法，一组动画中的所有动画会同时开始，一组动画完成后才会进行下一组动画。step 可以传入一个跟 uni.createAnimation() 一样的配置参数用于指定当前组动画的配置。

**示例代码**

<view :animation="animationData" style="background:red;height:100rpx;width:100rpx"></view>

export default{

data: {

animationData: {}

},

onShow: function(){

var animation = uni.createAnimation({

duration: 1000,

timingFunction: 'ease',

})

this.animation = animation

animation.scale(2,2).rotate(45).step()

this.animationData = animation.export()

setTimeout(function() {

animation.translate(30).step()

this.animationData = animation.export()

}.bind(this), 1000)

},

methods:{

rotateAndScale: function () {

// 旋转同时放大

this.animation.rotate(45).scale(2, 2).step()

this.animationData = this.animation.export()

},

rotateThenScale: function () {

// 先旋转后放大

this.animation.rotate(45).step()

this.animation.scale(2, 2).step()

this.animationData = this.animation.export()

},

rotateAndScaleThenTranslate: function () {

// 先旋转同时放大，然后平移

this.animation.rotate(45).scale(2, 2).step()

this.animation.translate(100, 100).step({ duration: 1000 })

this.animationData = this.animation.export()

}

}

}

## 滚动

### [uni.pageScrollTo(OBJECT)](https://uniapp.dcloud.io/api/ui/scroll?id=pagescrollto)

将页面滚动到目标位置。

**OBJECT参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| scrollTop | Number | 是 | 滚动到页面的目标位置（单位px） |
| duration | Number | 否 | 滚动动画的时长，默认300ms，单位 ms |

**示例**

uni.pageScrollTo({

scrollTop: 0,

duration: 300

});

## 窗口

### [uni.onWindowResize(CALLBACK)](https://uniapp.dcloud.io/api/ui/window?id=onwindowresize)

监听窗口尺寸变化事件

**1.6.0 新增**

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √(nvue不支持) | √ | √ | x | x | x | √ |

**CALLBACK 参数说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| size | Object | 变化后的窗口的大小，单位为 px ，{windowWidth,windowHeight} |

**代码示例**

uni.onWindowResize((res) => {

console.log('变化后的窗口宽度=' + res.size.windowWidth)

console.log('变化后的窗口高度=' + res.size.windowHeight)

})

**TIPS**

* 如App端设置软键盘弹出方式为adjustResize ，则在键盘弹出时，会触发此事件。
* 横竖屏切换时，会触发此事件。

### [uni.offWindowResize(CALLBACK)](https://uniapp.dcloud.io/api/ui/window?id=offwindowresize)

取消监听窗口尺寸变化事件

**1.6.0 新增**

平台差异说明

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| √ | √ | √ | x | x | x |

**代码示例**

uni.offWindowResize(() => {

console.log('取消监听窗口尺寸变化事件')

})

## 字体

### [uni.loadFontFace(Object object)](https://uniapp.dcloud.io/api/ui/font?id=loadfontface)

动态加载字体。

注意:

1. 引入中文字体，体积过大时会发生错误，建议抽离出部分中文，减少体积，或者用图片替代
2. 微信小程序端只支持网络字体，字体链接必须是https。App支持网络或本地的字体（本地字体需使用[平台绝对路径](http://www.html5plus.org/doc/zh_cn/io.html#plus.io.convertLocalFileSystemURL)）。
3. 微信小程序端字体链接必须是同源下的，或开启了cors支持，微信小程序的域名是servicewechat.com
4. canvas等原生组件不支持使用接口添加的字体
5. 工具里提示 Faild to load font可以忽略

**平台差异说明**

| **5+App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| 1.9.0+ [自定义组件编译模式](https://ask.dcloud.net.cn/article/35843) | 2.3.4+ | 基础库 2.1.0+ | x | x | x | x |

**参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| family | String |  | 是 | 定义的字体名称 |
| source | String |  | 是 | 字体资源的地址。建议格式为 TTF 和 WOFF，WOFF2 在低版本的iOS上会不兼容。 |
| desc | Object |  | 否 | 可选的字体描述符 |
| success | Function |  | 否 | 接口调用成功的回调函数 |
| fail | Function |  | 否 | 接口调用失败的回调函数 |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**Object.desc 的结构**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| style | String | normal | 否 | 字体样式，可选值为 normal / italic / oblique |
| weight | String | normal | 否 | 字体粗细，可选值为 normal / bold / 100 / 200../ 900 |
| variant | String | normal | 否 | 设置小型大写字母的字体显示文本，可选值为 normal / small-caps / inherit |

**代码示例**

uni.loadFontFace({

family: 'Bitstream Vera Serif Bold',

source: 'url("https://sungd.github.io/Pacifico.ttf")',

success() {

console.log('success')

}

})

**Tips**

插件市场有加载字体的例子：<https://ext.dcloud.net.cn/plugin?id=954>

## 下拉刷新

### [onPullDownRefresh](https://uniapp.dcloud.io/api/ui/pulldown?id=onpulldownrefresh)

在 js 中定义 onPullDownRefresh 处理函数（和onLoad等生命周期函数同级），监听该页面用户下拉刷新事件。

* 需要在 pages.json 里，找到的当前页面的pages节点，并在 style 选项中开启 enablePullDownRefresh。
* 当处理完数据刷新后，uni.stopPullDownRefresh 可以停止当前页面的下拉刷新。

### [uni.startPullDownRefresh(OBJECT)](https://uniapp.dcloud.io/api/ui/pulldown?id=startpulldownrefresh)

开始下拉刷新，调用后触发下拉刷新动画，效果与用户手动下拉刷新一致。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 接口调用结果 |

### [uni.stopPullDownRefresh()](https://uniapp.dcloud.io/api/ui/pulldown?id=stoppulldownrefresh)

停止当前页面下拉刷新。

**示例**

pages.json

{

"pages": [

{

"path": "pages/index/index",

"style": {

"navigationBarTitleText": "uni-app",

"enablePullDownRefresh": true

}

}

],

"globalStyle": {

"navigationBarTextStyle": "white",

"navigationBarBackgroundColor": "#0faeff",

"backgroundColor": "#fbf9fe"

}

}

index.vue

// 仅做示例，实际开发中延时根据需求来使用。

export default {

data: {

text: 'uni-app'

},

onLoad: function (options) {

setTimeout(function () {

console.log('start pulldown');

}, 1000);

uni.startPullDownRefresh();

},

onPullDownRefresh() {

console.log('refresh');

setTimeout(function () {

uni.stopPullDownRefresh();

}, 1000);

}

}

**注意**

* 支付宝小程序startPullDownRefresh在开发者工具里会提示暂未开放，请勿使用
* 支付宝小程序startPullDownRefresh请使用真机调试（非真机预览）
* 后续支付宝小程序开发工具更新可能会有所修改

### [FAQ](https://uniapp.dcloud.io/api/ui/pulldown?id=faq)

Q：如何暂时禁用掉下拉刷新，待需要的时候再重新开启？  
A：App 平台下可以处理此类场景，详细参考：[uni-app 中实现动态禁用/开启下拉刷新](https://ask.dcloud.net.cn/article/35134)

Q：自定义title如何让下拉刷新在title之下  
A：App和H5端使用circle方式的下拉刷新，设offset在title高度之下。hello uni-app的模板-导航栏中有示例。小程序端无法实现，除非放弃原生下拉刷新，自己模拟下拉刷新，插件市场有类似插件，但性能不如原生下拉刷新。

Q：如何自定义下拉刷新样式  
A：小程序端的原生下拉刷新样式是固定的；App端原生的下拉刷新有2种样式可选择，下拉漏出雪花和下拉circle圈。如果使用nvue，可以使用[refresh组件](https://uniapp.dcloud.io/component/refresh)自定义下拉刷新，都是原生渲染。如果想使用scroll-view在前端实现自定义下拉刷新，需要注意列表不可太长和太复杂，否则会有性能问题。[插件市场](https://ext.dcloud.net.cn/)搜索下拉刷新有示例。

## 节点信息

### uni.createSelectorQuery()

返回一个 SelectorQuery 对象实例。可以在这个实例上使用 select 等方法选择节点，并使用 boundingClientRect 等方法选择需要查询的信息。

**Tips:**

* 使用 uni.createSelectorQuery() 需要在生命周期 mounted 后进行调用。
* 自定义组件编译模式（默认模式），需要使用到 selectorQuery.in 方法。

### SelectorQuery

查询节点信息的对象

#### selectorQuery.in(component)

将选择器的选取范围更改为自定义组件 component 内，返回一个 SelectorQuery 对象实例。（初始时，选择器仅选取页面范围的节点，不会选取任何自定义组件中的节点）。

**代码示例**

const query = uni.createSelectorQuery().in(this);

query.select('#id').boundingClientRect(data => {

console.log("得到布局位置信息" + JSON.stringify(data));

console.log("节点离页面顶部的距离为" + data.top);

}).exec();

**注意**

* 支付宝小程序不支持in(component)，使用无效果

#### selectorQuery.select(selector)

在当前页面下选择第一个匹配选择器 selector 的节点，返回一个 NodesRef 对象实例，可以用于获取节点信息。

**selector 说明：**

selector 类似于 CSS 的选择器，但仅支持下列语法。

* ID选择器：#the-id
* class选择器（可以连续指定多个）：.a-class.another-class
* 子元素选择器：.the-parent > .the-child
* 后代选择器：.the-ancestor .the-descendant
* 跨自定义组件的后代选择器：.the-ancestor >>> .the-descendant
* 多选择器的并集：#a-node, .some-other-nodes

#### selectorQuery.selectAll(selector)

在当前页面下选择匹配选择器 selector 的所有节点，返回一个 NodesRef 对象实例，可以用于获取节点信息。

#### selectorQuery.selectViewport()

选择显示区域，可用于获取显示区域的尺寸、滚动位置等信息，返回一个 NodesRef 对象实例。

#### selectorQuery.exec(callback)

执行所有的请求。请求结果按请求次序构成数组，在callback的第一个参数中返回。

### NodesRef

用于获取节点信息的对象

#### nodesRef.fields(object,callback)

获取节点的相关信息。第一个参数是节点相关信息配置（必选）；第二参数是方法的回调函数，参数是指定的相关节点信息。

**object 参数说明**

| **字段名** | **类型** | **默认值** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| id | Boolean | false | 否 | 是否返回节点 id |  |
| dataset | Boolean | false | 否 | 是否返回节点 dataset | App、微信小程序、H5 |
| rect | Boolean | false | 否 | 是否返回节点布局位置（left right top bottom） |  |
| size | Boolean | false | 否 | 是否返回节点尺寸（width height） |  |
| scrollOffset | Boolean | false | 否 | 是否返回节点的 scrollLeft scrollTop，节点必须是 scroll-view 或者 viewport |  |
| properties | Array＜string＞ | [] | 否 | 指定属性名列表，返回节点对应属性名的当前属性值（只能获得组件文档中标注的常规属性值，id class style 和事件绑定的属性值不可获取） | 仅 App 和微信小程序支持 |
| computedStyle | Array＜string＞ | [] | 否 | 指定样式名列表，返回节点对应样式名的当前值 | 仅 App 和微信小程序支持 |
| context | Boolean | false | 否 | 是否返回节点对应的 Context 对象 | 仅 App 和微信小程序支持 |

#### nodesRef.boundingClientRect(callback)

添加节点的布局位置的查询请求。相对于显示区域，以像素为单位。其功能类似于 DOM 的 getBoundingClientRect。返回 NodesRef 对应的 SelectorQuery。

**callback 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| id | String | 节点的 ID |
| dataset | Object | 节点的 dataset |
| left | Number | 节点的左边界坐标 |
| right | Number | 节点的右边界坐标 |
| top | Number | 节点的上边界坐标 |
| bottom | Number | 节点的下边界坐标 |
| width | Number | 节点的宽度 |
| height | Number | 节点的高度 |

#### nodesRef.scrollOffset(callback)

添加节点的滚动位置查询请求。以像素为单位。节点必须是 scroll-view 或者 viewport。返回 NodesRef 对应的 SelectorQuery。

**callback 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| id | String | 节点的 ID |
| dataset | Object | 节点的 dataset |
| scrollLeft | Number | 节点的水平滚动位置 |
| scrollTop | Number | 节点的竖直滚动位置 |

#### nodesRef.context(callback)

添加节点的 Context 对象查询请求。支持 [VideoContext](https://uniapp.dcloud.io/api/media/video-context)、[CanvasContext](https://uniapp.dcloud.io/api/canvas/CanvasContext)、和 [MapContext](https://uniapp.dcloud.io/api/location/map) 等的获取。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | HBuilderX 2.4.7+ | √ | x | x | x | √ |

**callback 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| context | Object | 节点对应的 Context 对象 |

#### nodesRef.node(callback)

获取 Node 节点实例。目前支持 Canvas 的获取。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | x | x | √ |

**callback 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| node | Object | 节点对应的 Node 实例 |

**注意**

* 目前仅能用于canvas
* canvas需设置type="webgl"才能正常使用

#### 代码示例

uni.createSelectorQuery().selectViewport().scrollOffset(res => {

console.log("竖直滚动位置" + res.scrollTop);

}).exec();

let view = uni.createSelectorQuery().in(this).select(".test");

view.fields({

size: true,

scrollOffset: true

}, data => {

console.log("得到节点信息" + JSON.stringify(data));

console.log("节点的宽为" + data.width);

}).exec();

view.boundingClientRect(data => {

console.log("得到布局位置信息" + JSON.stringify(data));

console.log("节点离页面顶部的距离为" + data.top);

}).exec();

**注意**

* nvue 暂不支持 uni.createSelectorQuery，暂时使用下面的方案

<template>

<view class="wrapper">

<view ref="box" class="box">

<text class="info">Width: {{size.width}}</text>

<text class="info">Height: {{size.height}}</text>

<text class="info">Top: {{size.top}}</text>

<text class="info">Bottom: {{size.bottom}}</text>

<text class="info">Left: {{size.left}}</text>

<text class="info">Right: {{size.right}}</text>

</view>

</view>

</template>

<script>

// 注意平台差异

// #ifdef APP-NVUE

const dom = weex.requireModule('dom')

// #endif

export default {

data () {

return {

size: {

width: 0,

height: 0,

top: 0,

bottom: 0,

left: 0,

right: 0

}

}

},

onReady () {

const result = dom.getComponentRect(this.$refs.box, option => {

console.log('getComponentRect:', option)

this.size = option.size

})

console.log('return value:', result)

console.log('viewport:', dom.getComponentRect('viewport'))

}

}

</script>

## 节点布局相交状态

节点布局交叉状态 API 可用于监听两个或多个组件节点在布局位置上的相交状态。这一组API常常可以用于推断某些节点是否可以被用户看见、有多大比例可以被用户看见。

### [uni.createIntersectionObserver([this], [options])](https://uniapp.dcloud.io/api/ui/intersection-observer?id=createintersectionobserver)

创建并返回一个 IntersectionObserver 对象实例。

**this说明：**

自定义组件实例。**支付宝小程序不支持此参数，传入仅为抹平写法差异**

**options 的可选参数为：**

| **字段名** | **类型** | **说明** |
| --- | --- | --- |
| thresholds | Array | 一个数值数组，包含所有阈值。默认为 [0]。 |
| initialRatio | Number | 初始的相交比例，如果调用时检测到的相交比例与这个值不相等且达到阈值，则会触发一次监听器的回调函数。默认为 0。 |
| observeAll | Boolean | 是否同时观测多个参照节点（而非一个），如果设为 true，observe 的 targetSelector 将选中多个节点（注意：同时选中过多节点将影响渲染性能） |

### [IntersectionObserver 对象的方法列表](https://uniapp.dcloud.io/api/ui/intersection-observer?id=intersectionobserver-%e5%af%b9%e8%b1%a1%e7%9a%84%e6%96%b9%e6%b3%95%e5%88%97%e8%a1%a8)

| **方法** | **说明** |
| --- | --- |
| IntersectionObserver.relativeTo(selector,[margins]) | 使用选择器指定一个节点，作为参照区域之一。 |
| IntersectionObserver.relativeToViewport([margins]) | 指定页面显示区域作为参照区域之一 |
| IntersectionObserver.observe(selector,[callback]) | 指定目标节点并开始监听相交状态变化情况。回调函数 callback 包含一个参数 result |
| IntersectionObserver.disconnect() | 停止监听。回调函数将不再触发。 |

**margins 参数：** 用来扩展（或收缩）参照节点布局区域的边界。

| **属性** | **类型** | **默认值** | **是否必填** | **说明** |
| --- | --- | --- | --- | --- |
| left | number |  | 否 | 节点布局区域的左边界 |
| right | number |  | 否 | 节点布局区域的右边界 |
| top | number |  | 否 | 节点布局区域的上边界 |
| bottom | number |  | 否 | 节点布局区域的下边界 |

下面的示例代码中，如果目标节点 ".test" 进入 ".scroll" 区域以下 100px 时，就会触发回调函数。

uni.createIntersectionObserver(this).relativeTo('.scroll',{bottom: 100}).observe('.test', (res) => {

console.log(res);

})

**observe 回调函数 result 包含的字段**

| **字段名** | **类型** | **说明** |
| --- | --- | --- |
| intersectionRatio | Number | 相交比例 |
| intersectionRect | Object | 相交区域的边界，包含 left、right、top、bottom 四项 |
| boundingClientRect | Object | 目标节点布局区域的边界，包含 left、right、top、bottom 四项 |
| relativeRect | Object | 参照区域的边界，包含 left、right、top、bottom 四项 |
| time | Number | 相交检测时的时间戳 |

**Tips**

* 与页面显示区域的相交区域并不准确代表用户可见的区域，因为参与计算的区域是“布局区域”，布局区域可能会在绘制时被其他节点裁剪隐藏（如祖先节点中 overflow 样式为 hidden 的节点）或遮盖（如 fixed 定位的节点）。
* 节点交互状态 API 建议在 onReady 生命周期里监听，因为此 API 需要查找页面元素，onReady 时页面已经完成初次渲染，已经能查找到对应的元素。

### [代码示例](https://uniapp.dcloud.io/api/ui/intersection-observer?id=%e4%bb%a3%e7%a0%81%e7%a4%ba%e4%be%8b)

<template>

<view class="container">

<text>{{appear ? '小球出现' : '小球消失'}}</text>

<view class="page-section">

<scroll-view class="scroll-view" scroll-y>

<view class="scroll-area">

<text class="notice">向下滚动让小球出现</text>

<view class="ball"></view>

</view>

</scroll-view>

</view>

</view>

</template>

<script>

let observer = null;

export default {

data() {

return {

appear: false

}

},

onReady() {

observer = uni.createIntersectionObserver(this);

observer.relativeTo('.scroll-view').observe('.ball', (res) => {

if (res.intersectionRatio > 0 && !this.appear) {

this.appear = true;

} else if (!res.intersectionRatio > 0 && this.appear) {

this.appear = false;

}

})

},

onUnload() {

if (observer) {

observer.disconnect()

}

}

}

</script>

<style>

view,page {

display: flex;

flex-direction: column;

}

.scroll-view {

height: 400rpx;

background: #fff;

border: 1px solid #ccc;

box-sizing: border-box;

}

.scroll-area {

height: 1300rpx;

display: flex;

flex-direction: column;

align-items: center;

transition: .5s;

}

.notice {

margin-top: 150rpx;

margin: 150rpx 0 400rpx 0;

}

.ball {

width: 200rpx;

height: 200rpx;

background: #1AAD19;

border-radius: 50%;

}

</style>

## 自定义组件

#### [nextTick(function callback)](https://uniapp.dcloud.io/api/ui/nextTick?id=nexttick)

在小程序自定义组件，如wxcomponents中使用。延迟一部分操作到下一个时间片再执行。（类似于 setTimeout） 。其他平台无此概念。

* 微信小程序：[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/wx.nextTick.html)
* 百度小程序：[规范详情](https://smartprogram.baidu.com/docs/develop/api/custom_component/#swan-nextTick/)
* QQ小程序：[规范详情](https://q.qq.com/wiki/develop/miniprogram/API/interface/interface_nexttick.html#qq-nexttick)

## 菜单

#### [getMenuButtonBoundingClientRect()](https://uniapp.dcloud.io/api/ui/menuButton?id=getmenubuttonboundingclientrect)

在小程序平台，如果原生导航栏被隐藏，仍然在右上角会有一个悬浮按钮，微信下也被称为胶囊按钮。本API用于获取小程序下该菜单按钮的布局位置信息，方便开发者布局顶部内容时避开该按钮。

坐标信息以屏幕左上角为原点。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | √ | √ |

**返回值说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| width | number | 宽度，单位：px |
| height | number | 高度，单位：px |
| top | number | 上边界坐标，单位：px |
| right | number | 右边界坐标，单位：px |
| bottom | number | 下边界坐标，单位：px |
| left | number | 左边界坐标，单位：px |

**示例**

let menuButtonInfo = uni.getMenuButtonBoundingClientRect()

**注意**

* 支付宝小程序：其逻辑与微信小程序不同，它提供了菜单点击后按钮的自定义功能，可以选择显示那些系统按钮，[规范详情](https://docs.alipay.com/mini/api/optionmenuitem)

# 页面和窗体

## 页面

### [getCurrentPages()](https://uniapp.dcloud.io/api/window/window?id=getcurrentpages)

getCurrentPages() 函数用于获取当前页面栈的实例，以数组形式按栈的顺序给出，第一个元素为首页，最后一个元素为当前页面。

**注意：** getCurrentPages()仅用于展示页面栈的情况，请勿修改页面栈，以免造成页面状态错误。

每个页面实例的方法属性列表：

| **方法** | **描述** | **平台说明** |
| --- | --- | --- |
| page.$getAppWebview() | 获取当前页面的webview对象实例 | App |
| page.route | 获取当前页面的路由 |  |

Tips：

* navigateTo, redirectTo 只能打开非 tabBar 页面。
* switchTab 只能打开 tabBar 页面。
* reLaunch 可以打开任意页面。
* 页面底部的 tabBar 由页面决定，即只要是定义为 tabBar 的页面，底部都有 tabBar。
* 不能在 App.vue 里面进行页面跳转。

### [$getAppWebview()](https://uniapp.dcloud.io/api/window/window?id=getappwebview)

uni-app 在 getCurrentPages()获得的页面里内置了一个方法 $getAppWebview() 可以得到当前webview的对象实例，从而实现对 webview 更强大的控制。在 html5Plus 中，plus.webview具有强大的控制能力，可参考：[WebviewObject](http://www.html5plus.org/doc/zh_cn/webview.html#plus.webview.WebviewObject)。

但uni-app框架有自己的窗口管理机制，请不要自己创建和销毁webview，如有需求覆盖子窗体上去，请使用[原生子窗体subNvue](https://uniapp.dcloud.io/api/window/subNVues)。

**注意：此方法仅 App 支持**

**示例：**

获取当前页面 webview 的对象实例

export default {

data() {

return {

title: 'Hello'

}

},

onLoad() {

// #ifdef APP-PLUS

const currentWebview = this.$mp.page.$getAppWebview(); //此对象相当于html5plus里的plus.webview.currentWebview()。在uni-app里，直接使用plus.webview.currentWebview()无效

currentWebview.setBounce({position:{top:'100px'},changeoffset:{top:'0px'}}); //动态重设bounce效果

// #endif

}

}

获取指定页面 webview 的对象实例

getCurrentPages()可以得到所有页面对象，然后根据数组，可以取指定的页面webview对象

var pages = getCurrentPages();

var page = pages[pages.length - 1];

// #ifdef APP-PLUS

var currentWebview = page.$getAppWebview();

console.log(currentWebview.id);//获得当前webview的id

console.log(currentWebview.isVisible());//查询当前webview是否可见

);

// #endif

uni-app自带的web-view组件，是页面中新插入的一个子webview。获取该对象的方法见：<https://ask.dcloud.net.cn/article/35036>

## 页面通讯

**自 HBuilderX 2.0.0 起自定义组件编译模式支持，**[使用指南](https://ask.dcloud.net.cn/article/36010)

### [uni.$emit(eventName,OBJECT)](https://uniapp.dcloud.io/collocation/frame/communication?id=emit)

触发全局的自定事件。附加参数都会传给监听器回调。

| **属性** | **类型** | **描述** |
| --- | --- | --- |
| eventName | String | 事件名 |
| OBJECT | Object | 触发事件携带的附加参数 |

**代码示例**

uni.$emit('update',{msg:'页面更新'})

### [uni.$on(eventName,callback)](https://uniapp.dcloud.io/collocation/frame/communication?id=on)

监听全局的自定义事件。事件可以由 uni.$emit 触发，回调函数会接收所有传入事件触发函数的额外参数。

| **属性** | **类型** | **描述** |
| --- | --- | --- |
| eventName | String | 事件名 |
| callback | Function | 事件的回调函数 |

**代码示例**

uni.$on('update',function(data){

console.log('监听到事件来自 update ，携带参数 msg 为：' + data.msg);

})

### [uni.$once(eventName,callback)](https://uniapp.dcloud.io/collocation/frame/communication?id=once)

监听全局的自定义事件。事件可以由 uni.$emit 触发，但是只触发一次，在第一次触发之后移除监听器。

| **属性** | **类型** | **描述** |
| --- | --- | --- |
| eventName | String | 事件名 |
| callback | Function | 事件的回调函数 |

**代码示例**

uni.$once('update',function(data){

console.log('监听到事件来自 update ，携带参数 msg 为：' + data.msg);

})

### [uni.$off([eventName, callback])](https://uniapp.dcloud.io/collocation/frame/communication?id=off)

移除全局自定义事件监听器。

| **属性** | **类型** | **描述** |
| --- | --- | --- |
| eventName | Array＜String＞ | 事件名 |
| callback | Function | 事件的回调函数 |

**Tips**

* 如果没有提供参数，则移除所有的事件监听器；
* 如果只提供了事件，则移除该事件所有的监听器；
* 如果同时提供了事件与回调，则只移除这个回调的监听器；
* 提供的回调必须跟$on的回调为同一个才能移除这个回调的监听器；

**代码示例**

$emit、$on、$off常用于跨页面、跨组件通讯，这里为了方便演示放在同一个页面

<template>

<view class="content">

<view class="data">

<text>{{val}}</text>

</view>

<button type="primary" @click="comunicationOff">结束监听</button>

</view>

</template>

<script>

export default {

data() {

return {

val: 0

}

},

onLoad() {

setInterval(()=>{

uni.$emit('add', {

data: 2

})

},1000)

uni.$on('add', this.add)

},

methods: {

comunicationOff() {

uni.$off('add', this.add)

},

add(e) {

this.val += e.data

}

}

}

</script>

<style>

.content {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

.data {

text-align: center;

line-height: 40px;

margin-top: 40px;

}

button {

width: 200px;

margin: 20px 0;

}

</style>

**注意事项**

* uni.$emit、 uni.$on 、 uni.$once 、uni.$off 触发的事件都是 App 全局级别的，跨任意组件，页面，nvue，vue 等
* 使用时，注意及时销毁事件监听，比如，页面 onLoad 里边 uni.$on 注册监听，onUnload 里边 uni.$off 移除，或者一次性的事件，直接使用 uni.$once 监听

## SubNVue原子窗体

subNvue，是 vue 页面的原生子窗体，把weex渲染的原生界面当做 vue 页面的子窗体覆盖在页面上。它不是全屏页面，它给App平台vue页面中的层级覆盖和原生界面自定义提供了更强大和灵活的解决方案。它也不是组件，就是一个原生子窗体。

它的设计背景和用途详见：[subNVue 原生子窗体开发指南](https://ask.dcloud.net.cn/article/35948)

**subNvue 自1.9.10 起支持 ，仅支持 app 平台**

### [uni.getSubNVueById(subNvueId)](https://uniapp.dcloud.io/api/window/subNVues?id=app-getsubnvuebyid)

通过 ID 获取 subNVues 原生子窗体的实例。 [subNVues 配置](https://uniapp.dcloud.io/collocation/pages?id=app-subnvues)

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| subNvueId | String | 原生子窗体的 ID |

**代码示例**

const subNVue = uni.getSubNVueById('popup');

#### [返回值](https://uniapp.dcloud.io/api/window/subNVues?id=%e8%bf%94%e5%9b%9e%e5%80%bc)

返回一个 [subNVue](https://uniapp.dcloud.io/api/window/subNVues?id=subnvue) 对象

### [uni.getCurrentSubNVue()](https://uniapp.dcloud.io/api/window/subNVues?id=app-getcurrentsubnvue)

在一个subnvue窗体的nvue页面代码中，获取当前 subNVues 原生子窗体的实例。

**代码示例（注意执行在nvue页面中）**

const subNVue = uni.getCurrentSubNVue();

#### [返回值](https://uniapp.dcloud.io/api/window/subNVues?id=%e8%bf%94%e5%9b%9e%e5%80%bc-1)

返回一个 [subNVue](https://uniapp.dcloud.io/api/window/subNVues?id=subnvue) 对象

#### [subNVue 对象的方法列表](https://uniapp.dcloud.io/api/window/subNVues?id=subnvue)

| **方法** | **说明** |
| --- | --- |
| show | 显示原生子窗体 |
| hide | 隐藏原生子窗体 |
| setStyle | 设置原生子窗体的样式 |

### [subNVue.show (aniShow,duration,showedCB)](https://uniapp.dcloud.io/api/window/subNVues?id=subnvueshow)

显示原生子窗体

| **参数** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| aniShow | String | auto | 显示原生子窗体的动画效果 ，如果没有指定窗口动画类型，则使用默认值“auto”，即自动选择上一次显示窗口的动画效果，如果之前没有显示过，则使用“none”动画效果。[详见动画类型](https://uniapp.dcloud.io/api/window/subNVues?id=%e5%8a%a8%e7%94%bb%e7%b1%bb%e5%9e%8b) |
| duration | Number | 600 | 显示原生子窗体的动画持续时间，单位为ms，如果没有设置则使用默认窗口动画时间 600ms。 |
| showedCB | Function |  | 显示完成的回调函数，当指定原生子窗体显示动画执行完毕时触发回调函数，窗口无动画效果（如"none"动画效果）时也会触发此回调。 |

**代码示例**

subNVue.show('slide-in-left',200,()=>{

console.log('subNVue 原生子窗体显示成功');

})

### [subNVue.hide (aniShow,duration)](https://uniapp.dcloud.io/api/window/subNVues?id=subnvuehide)

隐藏原生子窗体

| **参数** | **类型** | **默认值** | **说明** |
| --- | --- | --- | --- |
| aniShow | String | auto | 隐藏原生子窗体的动画效果 ，如果没有指定窗口动画类型，则使用默认值“auto”，即自动选择上一次显示窗口的动画效果，如果之前没有显示过，则使用“none”动画效果。[详见动画类型](https://uniapp.dcloud.io/api/window/subNVues?id=%e5%8a%a8%e7%94%bb%e7%b1%bb%e5%9e%8b) |
| duration | Number | 600 | 隐藏原生子窗体的动画持续时间，单位为ms，如果没有设置则使用默认窗口动画时间 600ms。 |

**代码示例**

subNVue.hide('slide-out-left',200);

### [subNVue.setStyle (style)](https://uniapp.dcloud.io/api/window/subNVues?id=subnvuesetstyle)

设置原生子窗体的样式

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| style | Object | 原生子窗体的样式 |

**原生子窗体的样式**

| **属性** | **类型** | **默认值** | **描述** |
| --- | --- | --- | --- |
| position | String | absolute | 原生子窗体的排版位置，排版位置决定原生子窗体在父窗口中的定位方式。 可取值："static"，原生子窗体在页面中正常定位，如果页面存在滚动条则随窗口内容滚动；"absolute"，原生子窗体在页面中绝对定位，如果页面存在滚动条不随窗口内容滚动；"dock"，原生子窗体在页面中停靠，停靠的位置由dock属性值决定。 默认值为"absolute"。 |
| dock | String | bottom | 原生子窗体的停靠方式,仅当原生子窗体 "position" 属性值设置为 "dock" 时才生效，可取值："top"，原生子窗体停靠则页面顶部；"bottom"，原生子窗体停靠在页面底部；"right"，原生子窗体停靠在页面右侧；"left"，原生子窗体停靠在页面左侧。 默认值为"bottom"。 |
| mask | HexColor | rgba(0,0,0,0.5) | 原生子窗体的遮罩层,仅当原生子窗体 "type" 属性值设置为 "popup" 时才生效，可取值： rgba格式字符串，定义纯色遮罩层样式，如"rgba(0,0,0,0.5)"，表示黑色半透明； |
| width | String | 100% | 原生子窗体的宽度,支持百分比、像素值，默认为100%。未设置width属性值时，可同时设置left和right属性值改变窗口的默认宽度。 |
| height | String | 100% | 原生子窗体的高度,支持百分比、像素值，默认为100%。 当未设置height属性值时，优先通过top和bottom属性值来计算原生子窗体的高度。 |
| top | String | 0px | 原生子窗体垂直向下的偏移量，支持百分比、像素值，默认值为0px。 未设置top属性值时，优先通过bottom和height属性值来计算原生子窗体的top位置。 |
| bottom | String |  | 原生子窗体垂直向上偏移量,支持百分比、像素值，默认值无值（根据top和height属性值来自动计算）。 当同时设置了top和height值时，忽略此属性值； 当未设置height值时，可通过top和bottom属性值来确定原生子窗体的高度。 |
| left | String | 0px | 原生子窗体水平向左的偏移量，支持百分比、像素值，默认值为0px。 未设置left属性值时，优先通过right和width属性值来计算原生子窗体的left位置。 |
| right | String |  | 原生子窗体水平向右的偏移量，支持百分比、像素值，默认无值（根据left和width属性值来自动计算）。 当设置了left和width值时，忽略此属性值； 当未设置width值时，可通过left和bottom属性值来确定原生子窗体的宽度。 |
| margin | String |  | 原生子窗体的边距，用于定位原生子窗体的位置，支持auto，auto表示居中。若设置了left、right、top、bottom则对应的边距值失效。 |
| zindex | Number |  | 原生子窗体的窗口的堆叠顺序值，拥有更高堆叠顺序的窗口总是会处于堆叠顺序较低的窗口的前面，拥有相同堆叠顺序的窗口后调用show方法则在前面。 |

**代码示例**

subNVue.setStyle({

"position": "absolute", //除 popup 外，其他值域参考 5+ webview position 文档

"width": "50%",

"height": "50%",

"left":"20px",

"top":"100px"

})

### [subNVue.postMessage(OBJECT)](https://uniapp.dcloud.io/api/window/subNVues?id=subnvuepostmessage)

发送消息

**代码示例**

const subNVue = uni.getSubNVueById('subNvue');

subNvue.postMessage({

id:'1'

name:'subNvues'

})

### [subNVue.onMessage(CallBack)](https://uniapp.dcloud.io/api/window/subNVues?id=subnvueonmessage)

监听消息

**代码示例**

const subNVue = uni.getSubNVueById('subNvue');

subNvue.onMessage(function(data){

console.log('监听来自所属页面的 message' + JSON.stringify(data));

})

### [动画类型](https://uniapp.dcloud.io/api/window/subNVues?id=%e5%8a%a8%e7%94%bb%e7%b1%bb%e5%9e%8b)

显示动画与关闭动画，会有默认的对应规则。但是如果通过 API 原生子窗体的关闭动画类型，则不会使用默认的类型。

| **显示动画** | **关闭动画** | **显示动画描述（关闭动画与之相反）** |
| --- | --- | --- |
| slide-in-right | slide-out-right | 新窗体从右侧进入 |
| slide-in-left | slide-out-left | 新窗体从左侧进入 |
| slide-in-top | slide-out-top | 新窗体从顶部进入 |
| slide-in-bottom | slide-out-bottom | 新窗体从底部进入 |
| fade-in | fade-out | 新窗体从透明到不透明逐渐显示 |
| zoom-out | zoom-in | 新窗体从小到大缩放显示 |
| zoom-fade-out | zoom-fade-in | 新窗体从小到大逐渐放大并且从透明到不透明逐渐显示 |
| pop-in | pop-out | 新窗体从左侧进入，且老窗体被挤压而出 |
| none | none | 无动画 |

详细的窗口动画说明，请参考：

* 窗口显示的动画：[AnimationTypeShow](https://www.html5plus.org/doc/zh_cn/webview.html#plus.webview.AnimationTypeShow)
* 窗口关闭的动画：[AnimationTypeClose](https://www.html5plus.org/doc/zh_cn/webview.html#plus.webview.AnimationTypeClose)

**注意事项**

* 了解更多 subNVue 原生子窗体的用途，请详读[subNVues 原生子窗体开发指南](https://ask.dcloud.net.cn/article/35948)
* 在hello uni-app的接口-界面-原生子窗体中，有完整的subNVue示例，包括带渐变色的原生导航栏、可遮盖地图的侧滑菜单、可从顶部弹出的popup、可在视频上滚动的消息列表
* 每个 subNVue 页面都要在 pages.json 中注册。如果需要全局弹窗，也可以直接弹出nvue页面，参考这个[插件](https://ext.dcloud.net.cn/plugin?id=953)
* subNVue 比cover-view和plus.nativeObj.view更强大，也占用更多内存，为了保证更好的性能体验，一个vue页面不要加载太多 subNVue 子窗体，建议控制在三个以内

# 文件

#### [uni.saveFile(OBJECT)](https://uniapp.dcloud.io/api/file/file?id=savefile)

保存文件到本地。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**注意：saveFile 会把临时文件移动，因此调用成功后传入的 tempFilePath 将不可用。**

**OBJECT 参数说明：**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| tempFilePath | String | 是 | 需要保存的文件的临时路径 |
| success | Function | 否 | 返回文件的保存路径，res = {savedFilePath: '文件的保存路径'} |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **参数** | **说明** |
| --- | --- |
| savedFilePath | 文件的保存路径 |

**示例代码：**

uni.chooseImage({

success: function (res) {

var tempFilePaths = res.tempFilePaths;

uni.saveFile({

tempFilePath: tempFilePaths[0],

success: function (res) {

var savedFilePath = res.savedFilePath;

}

});

}

});

#### [uni.getSavedFileList(OBJECT)](https://uniapp.dcloud.io/api/file/file?id=getsavedfilelist)

获取本地已保存的文件列表。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**OBJECT 参数说明：**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | Function | 否 | 接口调用成功的回调函数，返回结果见 success 返回参数说明 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 接口调用结果 |
| fileList | Array<Object> | 文件列表 |

**fileList 中的项目说明：**

| **键** | **类型** | **说明** |
| --- | --- | --- |
| filePath | String | 文件的本地路径 |
| createTime | Number | 文件的保存时的时间戳，从 1970/01/01 08:00:00 到该时刻的秒数。 |
| size | String | 文件大小，以字节为单位。 |

**示例代码：**

uni.getSavedFileList({

success: function (res) {

console.log(res.fileList);

}

});

#### [uni.getSavedFileInfo(OBJECT)](https://uniapp.dcloud.io/api/file/file?id=getsavedfileinfo)

获取本地文件的文件信息。此接口只能用于获取已保存到本地的文件。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | x | √ |

**OBJECT 参数说明：**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| filePath | String | 是 | 文件路径 |
| success | Function | 否 | 接口调用成功的回调函数，返回结果见 success 返回参数说明 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数说明：**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 接口调用结果 |
| size | Number | 文件大小，以字节为单位。 |
| createTime | Number | 文件保存时的时间戳，从 1970/01/01 08:00:00 到该时刻的秒数。 |

**示例代码：**

uni.getSavedFileInfo({

filePath: 'unifile://somefile', //仅做示例用，非真正的文件路径

success: function (res) {

console.log(res.size);

console.log(res.createTime);

}

});

#### [uni.removeSavedFile(OBJECT)](https://uniapp.dcloud.io/api/file/file?id=removesavedfile)

删除本地存储的文件。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**OBJECT 参数说明：**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| filePath | String | 是 | 需要删除的文件路径 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例代码：**

uni.getSavedFileList({

success: function (res) {

if (res.fileList.length > 0) {

uni.removeSavedFile({

filePath: res.fileList[0].filePath,

complete: function (res) {

console.log(res);

}

});

}

}

});

#### [uni.getFileInfo(OBJECT)](https://uniapp.dcloud.io/api/file/file?id=getfileinfo)

获取文件信息

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**OBJECT 参数说明：**

| **参数名** | **类型** | **默认值** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| filePath | String |  | 是 | 本地文件路径 |  |
| digestAlgorithm | String | md5 | 否 | 计算文件摘要的算法，可取值 md5、sha1。 | 微信小程序 |
| success | Function |  | 否 | 接口调用成功的回调函数 |  |
| fail | Function |  | 否 | 接口调用失败的回调函数 |  |
| complete | Function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**success 返回参数说明：**

| **参数** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| errMsg | String | 接口调用结果 |  |
| size | Number | 文件大小，以字节为单位。 |  |
| digest | String | 按照传入的 digestAlgorithm 计算得出的的文件摘要 | 微信小程序 |

#### [uni.openDocument(OBJECT)](https://uniapp.dcloud.io/api/file/file?id=opendocument)

新开页面打开文档，支持格式：doc, xls, ppt, pdf, docx, xlsx, pptx。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**OBJECT 参数说明：**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| filePath | String | 是 | 文件路径，可通过 downFile 获得 |  |
| fileType | String | 否 | 文件类型，指定文件类型打开文件，有效值 doc, xls, ppt, pdf, docx, xlsx, pptx | 微信小程序 |
| success | String | 否 | 接口调用成功的回调函数 |  |
| fail | String | 否 | 接口调用失败的回调函数 | 微信小程序 |
| complete | String | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**示例代码：**

uni.downloadFile({

url: 'https://example.com/somefile.pdf',

success: function (res) {

var filePath = res.tempFilePath;

uni.openDocument({

filePath: filePath,

success: function (res) {

console.log('打开文档成功');

}

});

}

});

**平台差异**

| **平台** | **打开方式** |
| --- | --- |
| 小程序 | 在小程序的入口应用内打开 |
| App iOS | 在当前应用内打开 |
| App Android | 调用系统相关应用打开，无相关应用则不能打开 |
| H5 | 使用浏览器打开，当前浏览器不支持则不能打开 |

**Tips**

* App端，io操作还可以用更强大的plus.io API。[https://www.html5plus.org/doc/zh\_cn/io.html](https://www.html5plus.org/doc/zh_cn/io.html" \t "_blank)
* App端，打开各种格式的文件，如office、pdf等，还可以用更强大的三方插件，详见[插件市场](https://ext.dcloud.net.cn/search?q=pdf)
* 选择文件上传，uni-app有自带的api：[uni.chooseImage](https://uniapp.dcloud.io/api/media/image?id=chooseimage" \t "_blank)和[uni.chooseVideo](https://uniapp.dcloud.io/api/media/video?id=choosevideo" \t "_blank)。App端如需选择非媒体文件，可在插件市场搜索[文件选择](https://ext.dcloud.net.cn/search?q=%E6%96%87%E4%BB%B6%E9%80%89%E6%8B%A9)，其中Android端可以使用Native.js，无需原生插件，而iOS端需要原生插件。

# 绘画

#### [uni.createOffscreenCanvas()](https://uniapp.dcloud.io/api/canvas/createOffscreenCanvas?id=createoffscreencanvas)

创建离屏 canvas 实例

仅微信小程序平台支持，[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/wx.createOffscreenCanvas.html)

#### [uni.createCanvasContext(canvasId, this)](https://uniapp.dcloud.io/api/canvas/createCanvasContext?id=createcanvascontext)

**定义**

创建 canvas 绘图上下文（指定 canvasId）。在自定义组件下，第二个参数传入组件实例this，以操作组件内 <canvas/> 组件

**Tip:** 需要指定 canvasId，该绘图上下文只作用于对应的 <canvas/>

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| canvasId | String | 画布表示，传入定义在 <canvas/> 的 canvas-id |
| componentInstance | Object | 自定义组件实例 this ，表示在这个自定义组件下查找拥有 canvas-id 的 <canvas/> ，如果省略，则不在任何自定义组件内查找 |

**返回值**

[CanvasContext](https://uniapp.dcloud.io/api/canvas/CanvasContext)

#### [uni.canvasToTempFilePath(object, component)](https://uniapp.dcloud.io/api/canvas/canvasToTempFilePath?id=canvastotempfilepath)

把当前画布指定区域的内容导出生成指定大小的图片，并返回文件路径。在自定义组件下，第二个参数传入自定义组件实例，以操作组件内 <canvas> 组件。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | x（[CanvasContext.toTempFilePath](https://opendocs.alipay.com/mini/api/rod3ti)） | √ | √ | √ |

**object参数说明：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| x | Number | 否 | 画布x轴起点（默认0） |
| y | Number | 否 | 画布y轴起点（默认0） |
| width | Number | 否 | 画布宽度（默认为canvas宽度-x） |
| height | Number | 否 | 画布高度（默认为canvas高度-y） |
| destWidth | Number | 否 | 输出图片宽度（默认为 width \* 屏幕像素密度） |
| destHeight | Number | 否 | 输出图片高度（默认为 height \* 屏幕像素密度） |
| canvasId | String | 是 | 画布标识，传入 <canvas/> 的 canvas-id |
| fileType | String | 否 | 目标文件的类型，只支持 'jpg' 或 'png'。默认为 'png' |
| quality | Number | 否 | 图片的质量，取值范围为 (0, 1]，不在范围内时当作1.0处理 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例代码**

uni.canvasToTempFilePath({

x: 100,

y: 200,

width: 50,

height: 50,

destWidth: 100,

destHeight: 100,

canvasId: 'myCanvas',

success: function(res) {

// 在H5平台下，tempFilePath 为 base64

console.log(res.tempFilePath)

}

})

**Tips**

* H5端 Canvas 内绘制的图像需要支持跨域访问才能成功。

#### [uni.canvasPutImageData(OBJECT,this)](https://uniapp.dcloud.io/api/canvas/canvasPutImageData?id=canvasputimagedata)

将像素数据绘制到画布的方法，在自定义组件下，第二个参数传入自定义组件实例 this，以操作组件内 <canvas> 组件

**OBJECT参数说明：**

| **参数** | **类型** | **必填** | **说明** | **最低版本** |
| --- | --- | --- | --- | --- |
| canvasId | String | 是 | 画布标识，传入 <canvas /> 的 canvas-id |  |
| data | Uint8ClampedArray | 是 | 图像像素点数据，一维数组，每四项表示一个像素点的rgba |  |
| x | Number | 是 | 源图像数据在目标画布中的位置偏移量（x 轴方向的偏移量） |  |
| y | Number | 是 | 源图像数据在目标画布中的位置偏移量（y 轴方向的偏移量） |  |
| width | Number | 是 | 源图像数据矩形区域的宽度 |  |
| height | Number | 否 | 源图像数据矩形区域的高度 |  |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**示例代码**

const data = new Uint8ClampedArray([255, 0, 0, 1])

uni.canvasPutImageData({

canvasId: 'myCanvas',

x: 0,

y: 0,

width: 1,

data: data,

success(res) {}

})

#### [uni.canvasGetImageData(OBJECT,this)](https://uniapp.dcloud.io/api/canvas/canvasGetImageData?id=canvasgetimagedata)

返回一个数组，用来描述 canvas 区域隐含的像素数据，在自定义组件下，第二个参数传入自定义组件实例 this，以操作组件内 <canvas> 组件。

**OBJECT参数说明：**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| canvasId | String | 是 | 画布标识，传入 <canvas /> 的 canvas-id |
| x | Number | 是 | 将要被提取的图像数据矩形区域的左上角 x 坐标 |
| y | Number | 是 | 将要被提取的图像数据矩形区域的左上角 y 坐标 |
| width | Number | 是 | 将要被提取的图像数据矩形区域的宽度 |
| height | Number | 是 | 将要被提取的图像数据矩形区域的高度 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success回调返回参数：**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String |  |
| width | Number | 图像数据矩形的宽度 |
| height | Number | 图像数据矩形的高度 |
| data | Uint8ClampedArray | 图像像素点数据，一维数组，每四项表示一个像素点的rgba |

**示例代码**

uni.canvasGetImageData({

canvasId: 'myCanvas',

x: 0,

y: 0,

width: 100,

height: 100,

success(res) {

console.log(res.width) // 100

console.log(res.height) // 100

console.log(res.data instanceof Uint8ClampedArray) // true

console.log(res.data.length) // 100 \* 100 \* 4

}

})

### [CanvasContext](https://uniapp.dcloud.io/api/canvas/CanvasContext?id=canvascontextfillstyle-string)

**属性**

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | √ | √ |

App-nvue下如需使用canvas，暂未封装为uni API，可参考[文档](https://github.com/dcloudio/NvueCanvasDemo)使用。

#### CanvasContext.fillStyle string

填充颜色。用法同 [CanvasContext.setFillStyle()](https://uniapp.dcloud.io/CanvasContext?id=canvascontextsetfillstyle)。

#### CanvasContext.strokeStyle string

边框颜色。用法同 [CanvasContext.setStrokeStyle()](https://uniapp.dcloud.io/CanvasContext?id=canvascontextsetstrokestyle)。

#### CanvasContext.shadowOffsetX number

阴影相对于形状在水平方向的偏移

#### CanvasContext.shadowOffsetY number

阴影相对于形状在竖直方向的偏移

#### CanvasContext.shadowColor number

阴影的颜色

#### CanvasContext.shadowBlur number

阴影的模糊级别

#### CanvasContext.lineWidth number

线条的宽度。用法同 [CanvasContext.setLineWidth()](https://uniapp.dcloud.io/CanvasContext?id=canvascontextsetlinewidth)。

#### CanvasContext.lineCap number

线条的端点样式。用法同 [CanvasContext.setLineCap()](https://uniapp.dcloud.io/CanvasContext?id=canvascontextsetlinecap)。

#### CanvasContext.lineJoin number

线条的交点样式。用法同 [CanvasContext.setLineJoin()](https://uniapp.dcloud.io/CanvasContext?id=canvascontextsetlinejoin)。

#### CanvasContext.miterLimit number

最大斜接长度。用法同 [CanvasContext.setMiterLimit()](https://uniapp.dcloud.io/CanvasContext?id=canvascontextsetmiterlimit)。

#### CanvasContext.lineDashOffset number

虚线偏移量，初始值为0

#### CanvasContext.font string

当前字体样式的属性。符合 [CSS font 语法](https://developer.mozilla.org/zh-CN/docs/Web/CSS/font) 的 DOMString 字符串，至少需要提供字体大小和字体族名。默认值为 10px sans-serif。

#### CanvasContext.globalAlpha number

全局画笔透明度。范围 0-1，0 表示完全透明，1 表示完全不透明。

#### CanvasContext.globalCompositeOperation string

在绘制新形状时应用的合成操作的类型。目前安卓版本只适用于 fill 填充块的合成，用于 stroke 线段的合成效果都是 source-over。

目前支持的操作有

* App和H5端：source-over、destination-over、source-in、destination-in、source-out、destination-out、source-atop、destination-atop、lighter、darker、xor、copy
* 微信小程序安卓端：xor, source-over, source-atop, destination-out, lighter, overlay, darken, lighten, hard-light
* 微信小程序iOS端：xor, source-over, source-atop, destination-over, destination-out, lighter, multiply, overlay, darken, lighten, color-dodge, color-burn, hard-light, soft-light, difference, exclusion, saturation, luminosity

**方法**

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| √ | √ | √ | √ | √ | √ |

#### CanvasContext.arc

画一条弧线。创建一个圆可以用 arc() 方法指定其实弧度为0，终止弧度为 2 \* Math.PI。用 stroke() 或者 fill() 方法来在 canvas 中画弧线。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| x | Number | 圆的x坐标 |
| y | Number | 圆的y坐标 |
| r | Number | 圆的半径 |
| sAngle | Number | 起始弧度，单位弧度（在3点钟方向） |
| eAngle | Number | 终止弧度 |
| counterclockwise | Boolean | 可选。指定弧度的方向是逆时针还是顺时针。默认是false，即顺时针。 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

// Draw coordinates

ctx.arc(100, 75, 50, 0, 2 \* Math.PI)

ctx.setFillStyle('#EEEEEE')

ctx.fill()

ctx.beginPath()

ctx.moveTo(40, 75)

ctx.lineTo(160, 75)

ctx.moveTo(100, 15)

ctx.lineTo(100, 135)

ctx.setStrokeStyle('#AAAAAA')

ctx.stroke()

ctx.setFontSize(12)

ctx.setFillStyle('black')

ctx.fillText('0', 165, 78)

ctx.fillText('0.5\*PI', 83, 145)

ctx.fillText('1\*PI', 15, 78)

ctx.fillText('1.5\*PI', 83, 10)

// Draw points

ctx.beginPath()

ctx.arc(100, 75, 2, 0, 2 \* Math.PI)

ctx.setFillStyle('lightgreen')

ctx.fill()

ctx.beginPath()

ctx.arc(100, 25, 2, 0, 2 \* Math.PI)

ctx.setFillStyle('blue')

ctx.fill()

ctx.beginPath()

ctx.arc(150, 75, 2, 0, 2 \* Math.PI)

ctx.setFillStyle('red')

ctx.fill()

// Draw arc

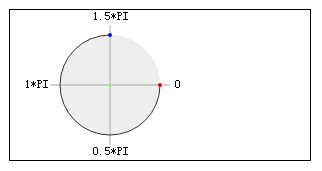
ctx.beginPath()

ctx.arc(100, 75, 50, 0, 1.5 \* Math.PI)

ctx.setStrokeStyle('#333333')

ctx.stroke()

ctx.draw()



针对 arc(100, 75, 50, 0, 1.5 \* Math.PI)的三个关键坐标如下：

* 绿色: 圆心 (100, 75)
* 红色: 起始弧度 (0)
* 蓝色: 终止弧度 (1.5 \* Math.PI)

#### CanvasContext.arcTo

根据控制点和半径绘制圆弧路径。

CanvasContext.arcTo(x1, y1, x2, y2, radius)

**参数**

| **属性值** | **类型** | **说明** |
| --- | --- | --- |
| x1 | Number | 第一个控制点的 x 轴坐标 |
| y1 | Number | 第一个控制点的 y 轴坐标 |
| x2 | Number | 第二个控制点的 x 轴坐标 |
| y2 | Number | 第二个控制点的 y 轴坐标 |
| radius | Number | 圆弧的半径 |

#### CanvasContext.beginPath

开始创建一个路径，需要调用fill或者stroke才会使用路径进行填充或描边。

**Tip:** 在最开始的时候相当于调用了一次 beginPath()。 **Tip:** 同一个路径内的多次setFillStyle()、``setStrokeStyle()、setLineWidth()```等设置，以最后一次设置为准。

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

// begin path

ctx.rect(10, 10, 100, 30)

ctx.setFillStyle('yellow')

ctx.fill()

// begin another path

ctx.beginPath()

ctx.rect(10, 40, 100, 30)

// only fill this rect, not in current path

ctx.setFillStyle('blue')

ctx.fillRect(10, 70, 100, 30)

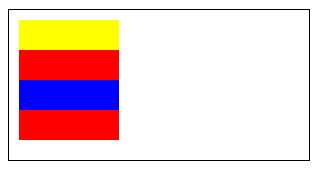
ctx.rect(10, 100, 100, 30)

// it will fill current path

ctx.setFillStyle('red')

ctx.fill()

ctx.draw()



#### CanvasContext.bezierCurveTo

创建三次方贝塞尔曲线路径。

**Tip:** 曲线的起始点为路径中前一个点。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| cp1x | Number | 第一个贝塞尔控制点的 x 坐标 |
| cp1y | Number | 第一个贝塞尔控制点的 y 坐标 |
| cp2x | Number | 第二个贝塞尔控制点的 x 坐标 |
| cp2y | Number | 第二个贝塞尔控制点的 y 坐标 |
| x | Number | 结束点的 x 坐标 |
| y | Number | 结束点的 y 坐标 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

// Draw points

ctx.beginPath()

ctx.arc(20, 20, 2, 0, 2 \* Math.PI)

ctx.setFillStyle('red')

ctx.fill()

ctx.beginPath()

ctx.arc(200, 20, 2, 0, 2 \* Math.PI)

ctx.setFillStyle('lightgreen')

ctx.fill()

ctx.beginPath()

ctx.arc(20, 100, 2, 0, 2 \* Math.PI)

ctx.arc(200, 100, 2, 0, 2 \* Math.PI)

ctx.setFillStyle('blue')

ctx.fill()

ctx.setFillStyle('black')

ctx.setFontSize(12)

// Draw guides

ctx.beginPath()

ctx.moveTo(20, 20)

ctx.lineTo(20, 100)

ctx.lineTo(150, 75)

ctx.moveTo(200, 20)

ctx.lineTo(200, 100)

ctx.lineTo(70, 75)

ctx.setStrokeStyle('#AAAAAA')

ctx.stroke()

// Draw quadratic curve

ctx.beginPath()

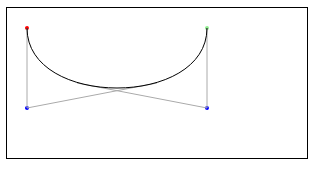
ctx.moveTo(20, 20)

ctx.bezierCurveTo(20, 100, 200, 100, 200, 20)

ctx.setStrokeStyle('black')

ctx.stroke()

ctx.draw()



针对 moveTo(20, 20)``bezierCurveTo(20, 100, 200, 100, 200, 20)``` 的三个关键坐标如下：

* 红色：起始点(20, 20)
* 蓝色：两个控制点(20, 100) (200, 100)
* 绿色：终止点(200, 20)

#### CanvasContext.clearRect

清除画布上在该矩形区域内的内容。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| x | Number | 矩形区域左上角的x坐标 |
| y | Number | 矩形区域左上角的y坐标 |
| width | Number | 矩形区域的宽度 |
| height | Number | 矩形区域的高度 |

**示例代码**

clearRect 并非画一个白色的矩形在地址区域，而是清空，为了有直观感受，对 canvas 加了一层背景色。

<canvas canvas-id="myCanvas" style="border: 1px solid; background: #123456;"/>

const ctx = uni.createCanvasContext('myCanvas')

ctx.setFillStyle('red')

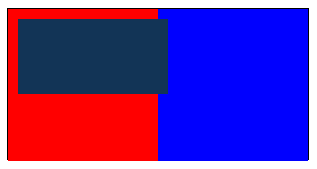
ctx.fillRect(0, 0, 150, 200)

ctx.setFillStyle('blue')

ctx.fillRect(150, 0, 150, 200)

ctx.clearRect(10, 10, 150, 75)

ctx.draw()



#### CanvasContext.clip

从原始画布中剪切任意形状和尺寸。一旦剪切了某个区域，则所有之后的绘图都会被限制在被剪切的区域内（不能访问画布上的其他区域）。可以在使用 clip() 方法前通过使用 save() 方法对当前画布区域进行保存，并在以后的任意时间对其进行恢复（通过 restore() 方法）。

**Tip:** 用 setFillStroke() 设置矩形线条的颜色，如果没设置默认是黑色。

**示例代码**

const context = uni.createCanvasContext('myCanvas')

uni.downloadFile({

url: 'https://img-cdn-qiniu.dcloud.net.cn/uniapp/images/uni@2x.png',

success: function (res) {

context.save()

context.beginPath()

context.arc(96, 96, 48, 0, 2 \* Math.PI)

context.clip()

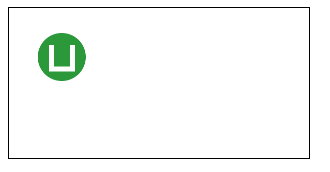
context.drawImage(res.tempFilePath, 48, 48)

context.restore()

context.draw()

}

})



#### CanvasContext.closePath

关闭一个路径。

**Tip:** 关闭路径会连接起点和终点。 **Tip:** 如果关闭路径后没有调用 fill() 或者 stroke() 并开启了新的路径，那之前的路径将不会被渲染。

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.moveTo(10, 10)

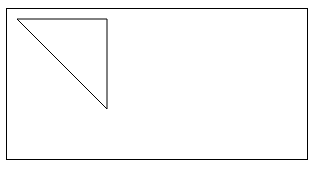
ctx.lineTo(100, 10)

ctx.lineTo(100, 100)

ctx.closePath()

ctx.stroke()

ctx.draw()



const ctx = uni.createCanvasContext('myCanvas')

// begin path

ctx.rect(10, 10, 100, 30)

ctx.closePath()

// begin another path

ctx.beginPath()

ctx.rect(10, 40, 100, 30)

// only fill this rect, not in current path

ctx.setFillStyle('blue')

ctx.fillRect(10, 70, 100, 30)

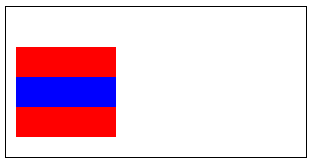
ctx.rect(10, 100, 100, 30)

// it will fill current path

ctx.setFillStyle('red')

ctx.fill()

ctx.draw()



#### CanvasContext.createCircularGradient

创建一个从圆心开始的渐变。返回的 [CanvasGradient](https://uniapp.dcloud.io/api/canvas/CanvasGradient) 对象，需要使用 CanvasGradient.addColorStop() 来指定渐变点，至少要两个。

**参数**

| **参数** | **类型** | **定义** |
| --- | --- | --- |
| x | Number | 圆心的x坐标 |
| y | Number | 圆心的y坐标 |
| r | Number | 圆的半径 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

// Create circular gradient

const grd = ctx.createCircularGradient(75, 50, 50)

grd.addColorStop(0, 'red')

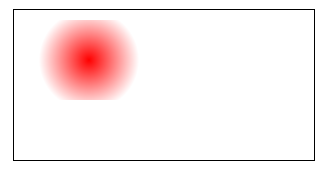
grd.addColorStop(1, 'white')

// Fill with gradient

ctx.setFillStyle(grd)

ctx.fillRect(10, 10, 150, 80)

ctx.draw()



#### CanvasContext.createLinearGradient

创建一个线性的渐变颜色。返回的 [CanvasGradient](https://uniapp.dcloud.io/api/canvas/CanvasGradient) 对象，需要使用 CanvasGradient.addColorStop() 来指定渐变点，至少要两个。

**参数**

| **参数** | **类型** | **定义** |
| --- | --- | --- |
| x0 | Number | 起点的x坐标 |
| y0 | Number | 起点的y坐标 |
| x1 | Number | 终点的x坐标 |
| y1 | Number | 终点的y坐标 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

// Create linear gradient

const grd = ctx.createLinearGradient(0, 0, 200, 0)

grd.addColorStop(0, 'red')

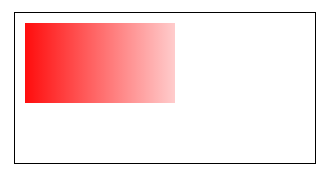
grd.addColorStop(1, 'white')

// Fill with gradient

ctx.setFillStyle(grd)

ctx.fillRect(10, 10, 150, 80)

ctx.draw()



#### CanvasContext.createPattern

对指定的图像创建模式的方法，可在指定的方向上重复元图像

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| image | String | 重复的图像源，仅支持包内路径和临时路径 |
| repetition | String | 指定如何重复图像，有效值有: repeat, repeat-x, repeat-y, no-repeat |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

const pattern = ctx.createPattern('/path/to/image', 'repeat-x')

ctx.fillStyle = pattern

ctx.fillRect(0, 0, 300, 150)

ctx.draw()

#### CanvasContext.draw

将之前在绘图上下文中的描述（路径、变形、样式）画到 canvas 中。

**参数**

| **参数** | **类型** | **说明** | **最低版本** |
| --- | --- | --- | --- |
| reserve | Boolean | 非必填。本次绘制是否接着上一次绘制，即reserve参数为false，则在本次调用drawCanvas绘制之前native层应先清空画布再继续绘制；若reserver参数为true，则保留当前画布上的内容，本次调用drawCanvas绘制的内容覆盖在上面，默认 false |  |
| callback | Function | 绘制完成后回调 | 1.7.0 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

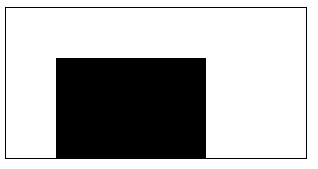
ctx.setFillStyle('red')

ctx.fillRect(10, 10, 150, 100)

ctx.draw()

ctx.fillRect(50, 50, 150, 100)

ctx.draw()



const ctx = uni.createCanvasContext('myCanvas')

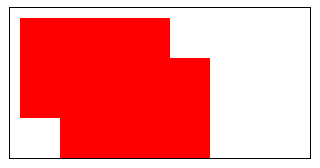
ctx.setFillStyle('red')

ctx.fillRect(10, 10, 150, 100)

ctx.draw()

ctx.fillRect(50, 50, 150, 100)

ctx.draw(true)



#### CanvasContext.drawImage

绘制图像到画布。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| imageResource | String | 所要绘制的图片资源 |
| dx | Number | 图像的左上角在目标canvas上 X 轴的位置 |
| dy | Number | 图像的左上角在目标canvas上 Y 轴的位置 |
| dWidth | Number | 在目标画布上绘制图像的宽度，允许对绘制的图像进行缩放 |
| dHeight | Number | 在目标画布上绘制图像的高度，允许对绘制的图像进行缩放 |
| sx | Number | 源图像的矩形选择框的左上角 X 坐标 |
| sy | Number | 源图像的矩形选择框的左上角 Y 坐标 |
| sWidth | Number | 源图像的矩形选择框的高度 |
| sHeight | Number | 源图像的矩形选择框的高度 |

有三个版本的写法：

* drawImage(dx, dy)
* drawImage(dx, dy, dWidth, dHeight)
* drawImage(sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight)

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

uni.chooseImage({

success: function(res){

ctx.drawImage(res.tempFilePaths[0], 0, 0, 150, 100)

ctx.draw()

}

})



#### CanvasContext.fill

对当前路径中的内容进行填充。默认的填充色为黑色。

**Tip:** 如果当前路径没有闭合，fill() 方法会将起点和终点进行连接，然后填充，详情见例一。

**Tip:** fill() 填充的的路径是从 beginPath() 开始计算，但是不会将 fillRect() 包含进去，详情见例二。

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

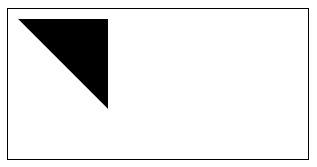
ctx.moveTo(10, 10)

ctx.lineTo(100, 10)

ctx.lineTo(100, 100)

ctx.fill()

ctx.draw()



const ctx = uni.createCanvasContext('myCanvas')

// begin path

ctx.rect(10, 10, 100, 30)

ctx.setFillStyle('yellow')

ctx.fill()

// begin another path

ctx.beginPath()

ctx.rect(10, 40, 100, 30)

// only fill this rect, not in current path

ctx.setFillStyle('blue')

ctx.fillRect(10, 70, 100, 30)

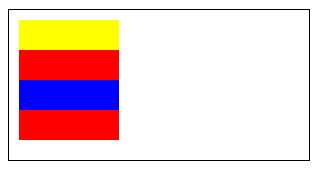
ctx.rect(10, 100, 100, 30)

// it will fill current path

ctx.setFillStyle('red')

ctx.fill()

ctx.draw()



#### CanvasContext.fillRect

填充一个矩形。

**Tip:** 用 setFillStyle() 设置矩形的填充色，如果没设置默认是黑色。

**参数**

| **参数** | **类型** | **定义** |
| --- | --- | --- |
| x | Number | 矩形路径左上角的x坐标 |
| y | Number | 矩形路径左上角的y坐标 |
| width | Number | 矩形路径的宽度 |
| height | Number | 矩形路径的高度 |

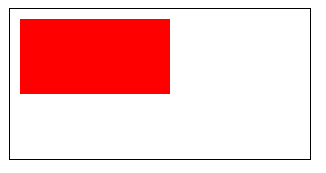
**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setFillStyle('red')

ctx.fillRect(10, 10, 150, 75)

ctx.draw()



#### CanvasContext.fillText

在画布上绘制被填充的文本。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| text | String | 在画布上输出的文本 |
| x | Number | 绘制文本的左上角x坐标位置 |
| y | Number | 绘制文本的左上角y坐标位置 |
| maxWidth | Number | 需要绘制的最大宽度，可选 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setFontSize(20)

ctx.fillText('Hello', 20, 20)

ctx.fillText('MINA', 100, 100)

ctx.draw()

#### CanvasContext.lineTo

增加一个新点，然后创建一条从上次指定点到目标点的线。

**Tip:** 用 stroke() 方法来画线条

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| x | Number | 目标位置的x坐标 |
| y | Number | 目标位置的y坐标 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

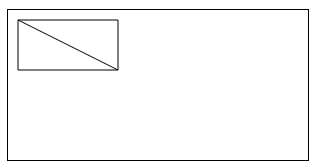
ctx.moveTo(10, 10)

ctx.rect(10, 10, 100, 50)

ctx.lineTo(110, 60)

ctx.stroke()

ctx.draw()



#### CanvasContext.measureText

测量文本尺寸信息，目前仅返回文本宽度。同步接口。(App端自定义组件编译模式暂时不可用)

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| text | String | 要测量的文本 |

**返回**

返回 TextMetrics 对象，结构如下：

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| width | Number | 文本的宽度 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.font = 'italic bold 20px cursive'

const metrics = ctx.measureText('Hello World')

console.log(metrics.width)

#### CanvasContext.moveTo

把路径移动到画布中的指定点，不创建线条。用 stroke() 方法来画线条。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| x | Number | 目标位置的x坐标 |
| y | Number | 目标位置的y坐标 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.moveTo(10, 10)

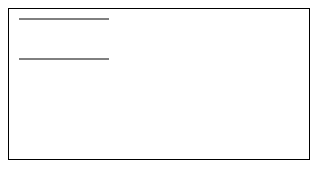
ctx.lineTo(100, 10)

ctx.moveTo(10, 50)

ctx.lineTo(100, 50)

ctx.stroke()

ctx.draw()



#### CanvasContext.quadraticCurveTo

创建二次贝塞尔曲线路径。曲线的起始点为路径中前一个点。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| cpx | Number | 贝塞尔控制点的x坐标 |
| cpy | Number | 贝塞尔控制点的y坐标 |
| x | Number | 结束点的x坐标 |
| y | Number | 结束点的y坐标 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

// Draw points

ctx.beginPath()

ctx.arc(20, 20, 2, 0, 2 \* Math.PI)

ctx.setFillStyle('red')

ctx.fill()

ctx.beginPath()

ctx.arc(200, 20, 2, 0, 2 \* Math.PI)

ctx.setFillStyle('lightgreen')

ctx.fill()

ctx.beginPath()

ctx.arc(20, 100, 2, 0, 2 \* Math.PI)

ctx.setFillStyle('blue')

ctx.fill()

ctx.setFillStyle('black')

ctx.setFontSize(12)

// Draw guides

ctx.beginPath()

ctx.moveTo(20, 20)

ctx.lineTo(20, 100)

ctx.lineTo(200, 20)

ctx.setStrokeStyle('#AAAAAA')

ctx.stroke()

// Draw quadratic curve

ctx.beginPath()

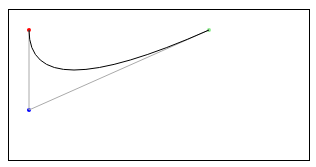
ctx.moveTo(20, 20)

ctx.quadraticCurveTo(20, 100, 200, 20)

ctx.setStrokeStyle('black')

ctx.stroke()

ctx.draw()



针对 moveTo(20, 20)``quadraticCurveTo(20, 100, 200, 20)``` 的三个关键坐标如下：

* 红色：起始点(20, 20)
* 蓝色：控制点(20, 100)
* 绿色：终止点(200, 20)

#### CanvasContext.rect

创建一个矩形。

**Tip:** 用 fill() 或者 stroke() 方法将矩形真正的画到 canvas 中。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| x | Number | 矩形路径左上角的x坐标 |
| y | Number | 矩形路径左上角的y坐标 |
| width | Number | 矩形路径的宽度 |
| height | Number | 矩形路径的高度 |

**示例代码**

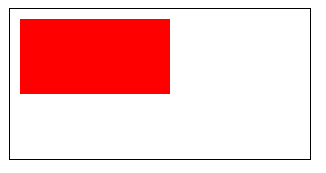
const ctx = uni.createCanvasContext('myCanvas')

ctx.rect(10, 10, 150, 75)

ctx.setFillStyle('red')

ctx.fill()

ctx.draw()



#### CanvasContext.restore

恢复之前保存的绘图上下文。

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

// save the default fill style

ctx.save()

ctx.setFillStyle('red')

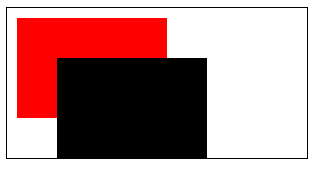
ctx.fillRect(10, 10, 150, 100)

// restore to the previous saved state

ctx.restore()

ctx.fillRect(50, 50, 150, 100)

ctx.draw()



#### CanvasContext.rotate

以原点为中心，原点可以用 translate方法修改。顺时针旋转当前坐标轴。多次调用rotate，旋转的角度会叠加。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| rotate | Number | 旋转角度，以弧度计(degrees \* Math.PI/180；degrees范围为0~360) |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.strokeRect(100, 10, 150, 100)

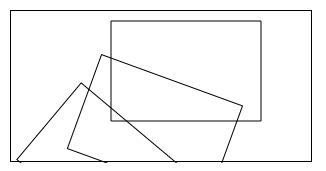
ctx.rotate(20 \* Math.PI / 180)

ctx.strokeRect(100, 10, 150, 100)

ctx.rotate(20 \* Math.PI / 180)

ctx.strokeRect(100, 10, 150, 100)

ctx.draw()



#### CanvasContext.save

保存当前的绘图上下文。

**示例代码**

const ctx = wx.createCanvasContext('myCanvas')

// save the default fill style

ctx.save()

ctx.setFillStyle('red')

ctx.fillRect(10, 10, 150, 100)

// restore to the previous saved state

ctx.restore()

ctx.fillRect(50, 50, 150, 100)

ctx.draw()

#### CanvasContext.scale

在调用scale方法后，之后创建的路径其横纵坐标会被缩放。多次调用scale，倍数会相乘。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| scaleWidth | Number | 横坐标缩放的倍数 (1 = 100%，0.5 = 50%，2 = 200%) |
| scaleHeight | Number | 纵坐标轴缩放的倍数 (1 = 100%，0.5 = 50%，2 = 200%) |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.strokeRect(10, 10, 25, 15)

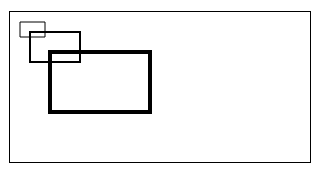
ctx.scale(2, 2)

ctx.strokeRect(10, 10, 25, 15)

ctx.scale(2, 2)

ctx.strokeRect(10, 10, 25, 15)

ctx.draw()



#### CanvasContext.setFillStyle

设置填充色，如果没有设置 fillStyle，默认颜色为 black。

**语法**

canvasContext.setFillStyle(color)

canvasContext.fillStyle = color

**参数**

| **参数** | **类型** | **定义** | **说明** |
| --- | --- | --- | --- |
| color | Color | Gradient Object | 填充色 |

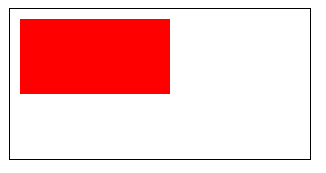
**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setFillStyle('red')

ctx.fillRect(10, 10, 150, 75)

ctx.draw()



#### CanvasContext.setFontSize

设置字体的字号。

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| fontSize | Number | 字体的字号 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setFontSize(20)

ctx.fillText('20', 20, 20)

ctx.setFontSize(30)

ctx.fillText('30', 40, 40)

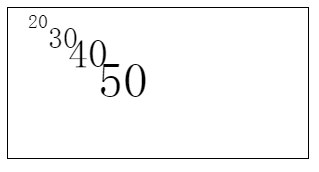
ctx.setFontSize(40)

ctx.fillText('40', 60, 60)

ctx.setFontSize(50)

ctx.fillText('50', 90, 90)

ctx.draw()



#### CanvasContext.setGlobalAlpha

设置全局画笔透明度。

**参数**

| **参数** | **类型** | **范围** | **说明** |
| --- | --- | --- | --- |
| alpha | Number | 0~1 | 透明度，0 表示完全透明，1 表示完全不透明 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setFillStyle('red')

ctx.fillRect(10, 10, 150, 100)

ctx.setGlobalAlpha(0.2)

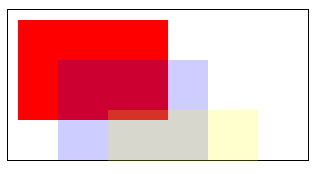
ctx.setFillStyle('blue')

ctx.fillRect(50, 50, 150, 100)

ctx.setFillStyle('yellow')

ctx.fillRect(100, 100, 150, 100)

ctx.draw()



#### CanvasContext.setLineCap

设置线条的端点样式。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| lineCap | String | 'butt'、'round'、'square' | 线条的结束端点样式 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.beginPath()

ctx.moveTo(10, 10)

ctx.lineTo(150, 10)

ctx.stroke()

ctx.beginPath()

ctx.setLineCap('butt')

ctx.setLineWidth(10)

ctx.moveTo(10, 30)

ctx.lineTo(150, 30)

ctx.stroke()

ctx.beginPath()

ctx.setLineCap('round')

ctx.setLineWidth(10)

ctx.moveTo(10, 50)

ctx.lineTo(150, 50)

ctx.stroke()

ctx.beginPath()

ctx.setLineCap('square')

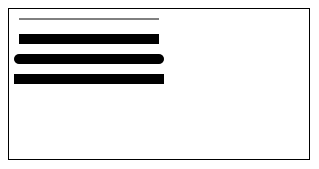
ctx.setLineWidth(10)

ctx.moveTo(10, 70)

ctx.lineTo(150, 70)

ctx.stroke()

ctx.draw()



#### CanvasContext.setLineDash

设置线条宽度。

**参数**

| **参数** | **类型** | **定义** |
| --- | --- | --- |
| pattern | Array | 一组描述交替绘制线段和间距（坐标空间单位）长度的数字 |
| offset | Number | 虚线偏移量 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setLineDash([10, 20], 5);

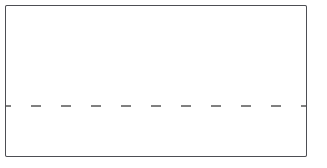
ctx.beginPath();

ctx.moveTo(0,100);

ctx.lineTo(400, 100);

ctx.stroke();

ctx.draw()



#### CanvasContext.setLineJoin

设置线条的交点样式。

**参数**

| **参数** | **类型** | **范围** | **说明** |
| --- | --- | --- | --- |
| lineJoin | String | 'bevel'、'round'、'miter' | 线条的结束交点样式 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.beginPath()

ctx.moveTo(10, 10)

ctx.lineTo(100, 50)

ctx.lineTo(10, 90)

ctx.stroke()

ctx.beginPath()

ctx.setLineJoin('bevel')

ctx.setLineWidth(10)

ctx.moveTo(50, 10)

ctx.lineTo(140, 50)

ctx.lineTo(50, 90)

ctx.stroke()

ctx.beginPath()

ctx.setLineJoin('round')

ctx.setLineWidth(10)

ctx.moveTo(90, 10)

ctx.lineTo(180, 50)

ctx.lineTo(90, 90)

ctx.stroke()

ctx.beginPath()

ctx.setLineJoin('miter')

ctx.setLineWidth(10)

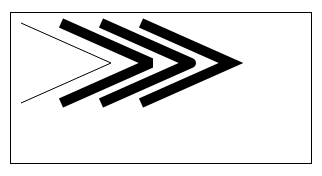
ctx.moveTo(130, 10)

ctx.lineTo(220, 50)

ctx.lineTo(130, 90)

ctx.stroke()

ctx.draw()



#### CanvasContext.setLineWidth

设置线条的宽度。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| lineWidth | Number | 线条的宽度(单位是px) |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.beginPath()

ctx.moveTo(10, 10)

ctx.lineTo(150, 10)

ctx.stroke()

ctx.beginPath()

ctx.setLineWidth(5)

ctx.moveTo(10, 30)

ctx.lineTo(150, 30)

ctx.stroke()

ctx.beginPath()

ctx.setLineWidth(10)

ctx.moveTo(10, 50)

ctx.lineTo(150, 50)

ctx.stroke()

ctx.beginPath()

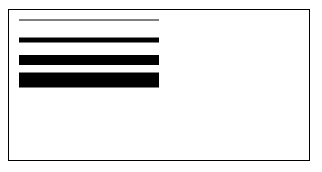
ctx.setLineWidth(15)

ctx.moveTo(10, 70)

ctx.lineTo(150, 70)

ctx.stroke()

ctx.draw()



#### CanvasContext.setMiterLimit

设置最大斜接长度，斜接长度指的是在两条线交汇处内角和外角之间的距离。 当 setLineJoin() 为 miter 时才有效。超过最大倾斜长度的，连接处将以 lineJoin 为 bevel 来显示。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| miterLimit | Number | 最大斜接长度 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.beginPath()

ctx.setLineWidth(10)

ctx.setLineJoin('miter')

ctx.setMiterLimit(1)

ctx.moveTo(10, 10)

ctx.lineTo(100, 50)

ctx.lineTo(10, 90)

ctx.stroke()

ctx.beginPath()

ctx.setLineWidth(10)

ctx.setLineJoin('miter')

ctx.setMiterLimit(2)

ctx.moveTo(50, 10)

ctx.lineTo(140, 50)

ctx.lineTo(50, 90)

ctx.stroke()

ctx.beginPath()

ctx.setLineWidth(10)

ctx.setLineJoin('miter')

ctx.setMiterLimit(3)

ctx.moveTo(90, 10)

ctx.lineTo(180, 50)

ctx.lineTo(90, 90)

ctx.stroke()

ctx.beginPath()

ctx.setLineWidth(10)

ctx.setLineJoin('miter')

ctx.setMiterLimit(4)

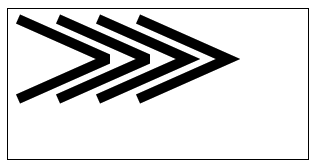
ctx.moveTo(130, 10)

ctx.lineTo(220, 50)

ctx.lineTo(130, 90)

ctx.stroke()

ctx.draw()



#### CanvasContext.setShadow

设置阴影样式。如果没有设置，offsetX 默认值为0， offsetY 默认值为0， blur 默认值为0，color 默认值为 black。

**参数**

| **参数** | **类型** | **定义** | **说明** |
| --- | --- | --- | --- |
| offsetX | Number |  | 阴影相对于形状在水平方向的偏移 |
| offsetY | Number |  | 阴影相对于形状在竖直方向的偏移 |
| blur | Number | 0~100 | 阴影的模糊级别，数值越大越模糊 |
| color | Color |  | 阴影的颜色 |

**示例代码**

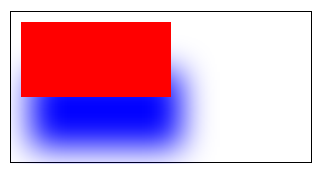
const ctx = uni.createCanvasContext('myCanvas')

ctx.setFillStyle('red')

ctx.setShadow(10, 50, 50, 'blue')

ctx.fillRect(10, 10, 150, 75)

ctx.draw()



#### CanvasContext.setStrokeStyle

设置边框颜色。如果没有设置 fillStyle，默认颜色为 black。

**参数**

| **参数** | **类型** | **定义** | **说明** |
| --- | --- | --- | --- |
| color | Color | Gradient Object | 填充色 |

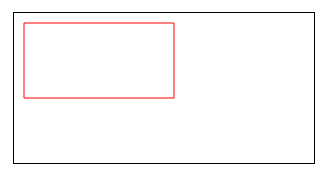
**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setStrokeStyle('red')

ctx.strokeRect(10, 10, 150, 75)

ctx.draw()



#### CanvasContext.setTextAlign

用于设置文字的对齐

| **参数** | **类型** | **定义** |
| --- | --- | --- |
| align | String | 可选值 'left'、'center'、'right' |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setStrokeStyle('red')

ctx.moveTo(150, 20)

ctx.lineTo(150, 170)

ctx.stroke()

ctx.setFontSize(15)

ctx.setTextAlign('left')

ctx.fillText('textAlign=left', 150, 60)

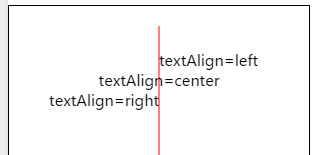
ctx.setTextAlign('center')

ctx.fillText('textAlign=center', 150, 80)

ctx.setTextAlign('right')

ctx.fillText('textAlign=right', 150, 100)

ctx.draw()



#### CanvasContext.setTextBaseline

用于设置文字的水平对齐

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| textBaseline | String | 可选值 'top'、'bottom'、'middle'、'normal' |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setStrokeStyle('red')

ctx.moveTo(5, 75)

ctx.lineTo(295, 75)

ctx.stroke()

ctx.setFontSize(20)

ctx.setTextBaseline('top')

ctx.fillText('top', 5, 75)

ctx.setTextBaseline('middle')

ctx.fillText('middle', 50, 75)

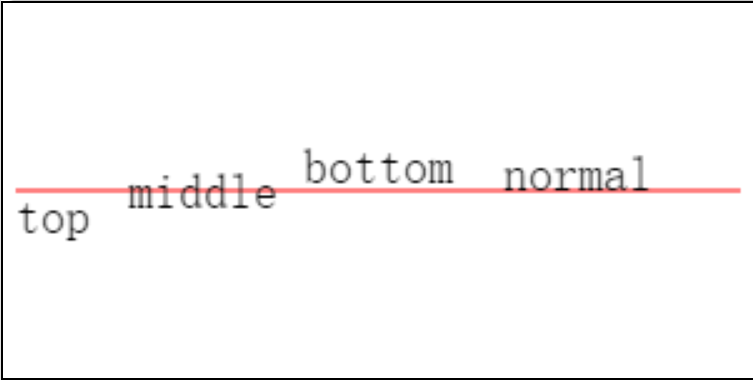
ctx.setTextBaseline('bottom')

ctx.fillText('bottom', 120, 75)

ctx.setTextBaseline('normal')

ctx.fillText('normal', 200, 75)

ctx.draw()



#### CanvasContext.setTransform

使用矩阵重新设置（覆盖）当前变换的方法

**语法**

canvasContext.setTransform(scaleX, skewX, skewY, scaleY, translateX, translateY)

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| scaleX | Number | 水平缩放 |
| skewX | Number | 水平倾斜 |
| skewY | Number | 垂直倾斜 |
| scaleY | Number | 垂直缩放 |
| translateX | Number | 水平移动 |
| translateY | Number | 垂直移动 |

#### CanvasContext.stroke

画出当前路径的边框。默认颜色色为黑色。

**Tip:** stroke() 描绘的的路径是从 beginPath() 开始计算，但是不会将 strokeRect() 包含进去，详情见例二。

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

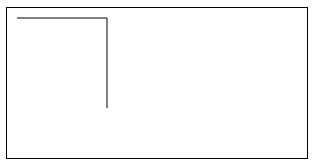
ctx.moveTo(10, 10)

ctx.lineTo(100, 10)

ctx.lineTo(100, 100)

ctx.stroke()

ctx.draw()



const ctx = uni.createCanvasContext('myCanvas')

// begin path

ctx.rect(10, 10, 100, 30)

ctx.setStrokeStyle('yellow')

ctx.stroke()

// begin another path

ctx.beginPath()

ctx.rect(10, 40, 100, 30)

// only stoke this rect, not in current path

ctx.setStrokeStyle('blue')

ctx.strokeRect(10, 70, 100, 30)

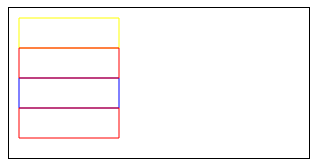
ctx.rect(10, 100, 100, 30)

// it will stroke current path

ctx.setStrokeStyle('red')

ctx.stroke()

ctx.draw()



#### CanvasContext.strokeRect

画一个矩形(非填充)。用 setFillStroke() 设置边框颜色，如果没设置默认是黑色。

**参数**

| **参数** | **类型** | **范围** | **说明** |
| --- | --- | --- | --- |
| x | Number |  | 矩形路径左上角的x坐标 |
| y | Number |  | 矩形路径左上角的y坐标 |
| width | Number |  | 矩形路径的宽度 |
| height | Number |  | 矩形路径的高度 |

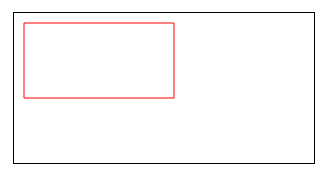
**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.setStrokeStyle('red')

ctx.strokeRect(10, 10, 150, 75)

ctx.draw()



#### CanvasContext.strokeText

给定的 (x, y) 位置绘制文本描边的方法

**语法**

canvasContext.strokeText(text, x, y, maxWidth)

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| text | String | 要绘制的文本 |
| x | Number | 文本起始点的 x 轴坐标 |
| y | Number | 文本起始点的 y 轴坐标 |
| maxWidth | Number | 需要绘制的最大宽度，可选 |

#### CanvasContext.transform

使用矩阵多次叠加当前变换的方法。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| scaleX | Number | 水平缩放 |
| skewX | Number | 水平倾斜 |
| skewY | Number | 垂直倾斜 |
| scaleY | Number | 垂直缩放 |
| translateX | Number | 水平移动 |
| translateY | Number | 垂直移动 |

#### CanvasContext.translate

对当前坐标系的原点(0, 0)进行变换，默认的坐标系原点为页面左上角。

**参数**

| **参数** | **类型** | **说明** |
| --- | --- | --- |
| x | Number | 水平坐标平移量 |
| y | Number | 竖直坐标平移量 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

ctx.strokeRect(10, 10, 150, 100)

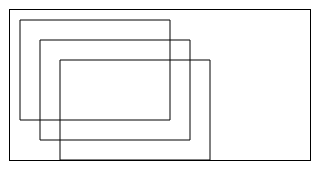
ctx.translate(20, 20)

ctx.strokeRect(10, 10, 150, 100)

ctx.translate(20, 20)

ctx.strokeRect(10, 10, 150, 100)

ctx.draw()



## CanvasGradient

### [CanvasGradient.addColorStop(stop,color)](https://uniapp.dcloud.io/api/canvas/CanvasGradient?id=canvasgradientaddcolorstop)

创建一个颜色的渐变点。

* 小于最小 stop 的部分会按最小 stop 的 color 来渲染，大于最大 stop 的部分会按最大 stop 的 color 来渲染。
* 需要使用 addColorStop() 来指定渐变点，至少要两个。

#### [参数](https://uniapp.dcloud.io/api/canvas/CanvasGradient?id=%e5%8f%82%e6%95%b0)

| **参数** | **类型** | **定义** |
| --- | --- | --- |
| stop | Number(0-1) | 表示渐变点在起点和终点中的位置 |
| color | Color | 渐变点的颜色 |

**示例代码**

const ctx = uni.createCanvasContext('myCanvas')

// Create circular gradient

const grd = ctx.createLinearGradient(30, 10, 120, 10)

grd.addColorStop(0, 'red')

grd.addColorStop(0.16, 'orange')

grd.addColorStop(0.33, 'yellow')

grd.addColorStop(0.5, 'green')

grd.addColorStop(0.66, 'cyan')

grd.addColorStop(0.83, 'blue')

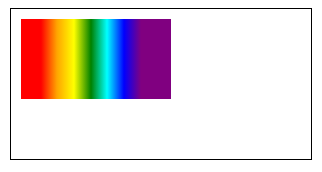
grd.addColorStop(1, 'purple')

// Fill with gradient

ctx.setFillStyle(grd)

ctx.fillRect(10, 10, 150, 80)

ctx.draw()



# 广告

### [激励视频广告](https://uniapp.dcloud.io/api/ad/rewarded-video-ad?id=%e6%bf%80%e5%8a%b1%e8%a7%86%e9%a2%91%e5%b9%bf%e5%91%8a)

* app端的广告源由腾讯广点通、头条穿山甲广告联盟提供，DCloud负责聚合
* 小程序端的广告由小程序平台提供

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √（2.5.11+） | x | √ | x | x | x | x |

**开通配置广告**

开通广告步骤：

1. 开通广告 需在广告平台后台操作：
   * App平台：<https://uniad.dcloud.net.cn/>
   * 小程序平台：在各自的小程序管理后台操作。
2. 申请广告位id 在各位后台申请广告位id
3. App端打包后生效，打包时必须选择要集成的广告SDK（穿山甲、广点通）。

激励视频广告组件是由客户端原生的图片、文本、视频控件组成的，层级最高，会覆盖在普通组件上。

### [广告创建](https://uniapp.dcloud.io/api/ad/rewarded-video-ad?id=%e5%b9%bf%e5%91%8a%e5%88%9b%e5%bb%ba)

开发者可以调用 uni.createRewardedVideoAd 创建激励视频广告组件。该方法返回的是一个单例，该实例仅对当前页面有效，不允许跨页面使用。

激励视频广告组件默认是隐藏的，因此可以提前创建，以提前初始化组件。开发者可以在页面的 onReady 事件回调中创建广告实例，并在该页面的生命周期内重复调用该广告实例。

<script>

let rewardedVideoAd = null;

export default {

data() {

return {

title: 'createRewardedVideoAd'

}

},

onReady() {

if(uni.createRewardedVideoAd) {

rewardedVideoAd = uni.createRewardedVideoAd({ adpid: 'xxxx' })

rewardedVideoAd.onLoad(() => {

console.log('onLoad event')

})

rewardedVideoAd.onError((err) => {

console.log('onError event', err)

})

rewardedVideoAd.onClose((res) => {

console.log('onClose event', res)

})

}

},

methods: {

}

}

</script>

为避免滥用广告资源，目前每个用户每天可观看激励式视频广告的次数有限，建议展示广告按钮前先判断广告是否拉取成功。

### [显示/隐藏](https://uniapp.dcloud.io/api/ad/rewarded-video-ad?id=%e6%98%be%e7%a4%ba%e9%9a%90%e8%97%8f)

激励视频广告组件默认是隐藏的，在用户主动触发广告后，开发者需要调用 RewardedVideoAd.show() 进行显示。

rewardedVideoAd.show()

只有在用户点击激励视频广告组件上的 关闭广告 按钮时，广告才会关闭。开发者不可控制激励视频广告组件的隐藏。

### [广告拉取成功与失败](https://uniapp.dcloud.io/api/ad/rewarded-video-ad?id=%e5%b9%bf%e5%91%8a%e6%8b%89%e5%8f%96%e6%88%90%e5%8a%9f%e4%b8%8e%e5%a4%b1%e8%b4%a5)

激励视频广告组件是自动拉取广告并进行更新的。在组件创建后会拉取一次广告，用户点击 关闭广告 后会去拉取下一条广告。

如果拉取成功，通过 RewardedVideoAd.onLoad() 注册的回调函数会执行，RewardedVideoAd.show() 返回的 Promise 也会是一个 resolved Promise。两者的回调函数中都没有参数传递。

rewardedVideoAd.onLoad(() => {

console.log('激励视频 广告加载成功')

})

rewardedVideoAd.show()

.then(() => console.log('激励视频 广告显示'))

如果拉取失败，通过 RewardedVideoAd.onError() 注册的回调函数会执行，回调函数的参数是一个包含错误信息的对象。常见异常错误参考文档

rewardedVideoAd.onError(err => {

console.log(err)

})

RewardedVideoAd.show() 返回的 Promise 也会是一个 rejected Promise。

rewardedVideoAd.show()

.catch(err => console.log(err))

### [拉取失败，重新拉取](https://uniapp.dcloud.io/api/ad/rewarded-video-ad?id=%e6%8b%89%e5%8f%96%e5%a4%b1%e8%b4%a5%ef%bc%8c%e9%87%8d%e6%96%b0%e6%8b%89%e5%8f%96)

如果组件的某次自动拉取失败，那么之后调用的 show() 将会被 reject。此时可以调用 RewardedVideoAd.load() 手动重新拉取广告。

rewardedVideoAd.show()

.catch(() => {

rewardedVideoAd.load()

.then(() => rewardedVideoAd.show())

.catch(err => {

console.log('激励视频 广告显示失败')

})

})

如果组件的自动拉取是成功的，那么调用 load() 方法会直接返回一个 resolved Promise，而不会去拉取广告。

rewardedVideoAd.load()

.then(() => rewardedVideoAd.show())

### [监听用户关闭广告](https://uniapp.dcloud.io/api/ad/rewarded-video-ad?id=%e7%9b%91%e5%90%ac%e7%94%a8%e6%88%b7%e5%85%b3%e9%97%ad%e5%b9%bf%e5%91%8a)

只有在用户点击激励视频广告组件上的 关闭广告 按钮时，广告才会关闭。这个事件可以通过 RewardedVideoAd.onClose() 监听。

RewardedVideoAd.onClose() 的回调函数会传入一个参数 res，res.isEnded 描述广告被关闭时的状态。

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| isEnded | boolean | 视频是否是在用户完整观看的情况下被关闭的，true 表示用户是在视频播放完以后关闭的视频，false 表示用户在视频播放过程中关闭了视频 |

开发者需要根据 res.isEnded 判断是否视频是否播放结束、可以向用户下发奖励。

rewardedVideoAd.onClose(res => {

// 用户点击了【关闭广告】按钮

if (res && res.isEnded) {

// 正常播放结束

} else {

// 播放中途退出

}

})

**注意事项**

多次调用 RewardedVideoAd.onLoad()、RewardedVideoAd.onError()、RewardedVideoAd.onClose() 等方法监听广告事件会产生多次事件回调，建议在创建广告后监听一次即可，或者先取消原有的监听事件再重新监听。

**AD组件** 文档地址：<https://uniapp.dcloud.io/component/ad>

# 第三方服务

## 获取服务供应商

### [uni.getProvider(OBJECT)](https://uniapp.dcloud.io/api/plugins/provider?id=getprovider)

获取服务供应商。

在App平台，可用的服务商，是打包环境中配置的服务商，与手机端是否安装了该服务商的App没有关系。

云打包在manifest中配置相关模块和SDK信息，离线打包在原生工程中配置。某个服务商配置被打包进去，运行时就能得到相应的服务供应商。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| service | String | 是 | 服务类型，可取值见下面说明。 |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**service 值说明**

| **值** | **说明** |
| --- | --- |
| oauth | 授权登录 |
| share | 分享 |
| payment | 支付 |
| push | 推送 |

**success 返回参数说明**

| **参数名** | **类型** | **说明** |
| --- | --- | --- |
| service | String | 服务类型 |
| provider | Array | 得到的服务供应商 |

**provider 在不同服务类型下可能的取值说明**

| **service** | **provider** | **说明** | **备注** |
| --- | --- | --- | --- |
| oauth | weixin | 微信登录 |  |
|  | qq | QQ登录 |  |
|  | sinaweibo | 新浪微博登录 |  |
|  | xiaomi | 小米登录 |  |
|  | apple | [Apple登录](https://ask.dcloud.net.cn/article/36651) | 仅iOS13支持，HBuilderX 2.4.7+ |
| share | sinaweibo | 新浪微博分享 |  |
|  | qq | 分享到QQ好友 |  |
|  | weixin | 分享微信消息、朋友圈及微信小程序 |  |
| payment | alipay | 支付宝支付 |  |
|  | wxpay | 微信支付 |  |
|  | baidu | 百度收银台 |  |
|  | appleiap | 苹果应用内支付 | iOS 应用打包后可获取 |
| push | unipush | [UniPush](https://ask.dcloud.net.cn/article/35622) | 推送服务是三选一，只会获取到一个供应商。 |
|  | igexin | 个推 | 填写配置并打包后可以获取，仅为向下兼容而保留，不再推荐使用 |
|  | mipush | 小米推送 | 填写配置并打包后可以获取，仅为向下兼容而保留，不再推荐使用 |

**注意事项**

* 自 HBuilderX 1.7.3 起，HBuilder 基座的推送供应商为 UniPush 服务。

**代码**

uni.getProvider({

service: 'oauth',

success: function (res) {

console.log(res.provider)

if (~res.provider.indexOf('qq')) {

uni.login({

provider: 'qq',

success: function (loginRes) {

console.log(JSON.stringify(loginRes));

}

});

}

}

});

## 登录

### [uni.login(OBJECT)](https://uniapp.dcloud.io/api/plugins/login?id=login)

登录

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

H5平台登陆注意事项：

* 微信内嵌浏览器运行H5版时，可通过js sdk实现微信登陆，需要引入一个单独的js，[详见](https://ask.dcloud.net.cn/article/35380" \t "_blank)
* 普通浏览器上实现微信登陆，并非开放API，需要向微信申请，仅个别开发者有此权限
* H5平台的其他登陆，比如QQ登陆、微博登陆，uni-app未封装，请在条件编译里按普通H5写法编写。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| provider | String | 否 | 登录服务提供商，通过 [uni.getProvider](https://uniapp.dcloud.io/api/plugins/provider) 获取，如果不设置则弹出登录列表选择界面 |  |
| scopes | String/Array | 见平台差异说明 | 授权类型，默认 auth\_base。支持 auth\_base（静默授权）/ auth\_user（主动授权） / auth\_zhima（芝麻信用） | 支付宝小程序 |
| timeout | Number | 否 | 超时时间，单位ms | 微信小程序、百度小程序 |
| success | Function | 否 | 接口调用成功的回调 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**success 返回参数说明**

| **参数名** | **说明** |
| --- | --- |
| authResult | 登录服务商提供的登录信息，服务商不同返回的结果不完全相同 |
| errMsg | 描述信息 |

各个平台的登录流程存在差异，详细请参考相关平台的文档说明：

* [微信小程序登录](https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/login.html)
* [支付宝小程序用户授权](https://docs.alipay.com/mini/introduce/authcode)
* [百度小程序登录](https://smartprogram.baidu.com/docs/develop/api/open_log/#%E6%8E%88%E6%9D%83%E6%B5%81%E7%A8%8B%E8%AF%B4%E6%98%8E/)
* [头条小程序登录](https://developer.toutiao.com/dev/cn/mini-app/develop/open-capacity/log-in/login)
* [Apple登录、苹果登录、Sign in with Apple](https://ask.dcloud.net.cn/article/36651)

**示例**

uni.login({

provider: 'weixin',

success: function (loginRes) {

console.log(loginRes.authResult);

}

});

### [uni.checkSession](https://uniapp.dcloud.io/api/plugins/login?id=unichecksession)

检查登录状态是否过期

**1.6.0 新增**

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | √ | √ |

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | function | 否 | 接口调用成功的回调函数 |
| fail | function | 否 | 接口调用失败的回调函数 |
| complete | function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

### [uni.getUserInfo(OBJECT)](https://uniapp.dcloud.io/api/plugins/login?id=getuserinfo)

获取用户信息。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | x | √ | √ | √ | √ | √ |

**注意：** 微信小程序端，在用户未授权过的情况下调用此接口，不会出现授权弹窗，会直接进入 fail 回调（详见[《微信小程序公告》](https://developers.weixin.qq.com/community/develop/doc/0000a26e1aca6012e896a517556c01)）。在用户已授权的情况下调用此接口，可成功获取用户信息。

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| provider | String | 否 | 登录服务提供商，通过 uni.getProvider 获取 |  |
| withCredentials | Boolean | 否 | 是否带上登录态信息。 | 微信小程序、头条小程序 |
| lang | String | 否 | 指定返回用户信息的语言，默认为 en。更多值请参考下面的说明。 | 微信小程序 |
| timeout | Number | 否 | 超时时间，单位 ms。 | 微信小程序 |
| success | Function | 否 | 接口调用成功的回调 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**lang 值说明**

| **值** | **说明** |
| --- | --- |
| zh\_CN | 简体中文 |
| zh\_TW | 繁体中文 |
| en | 英文 |

**注意：**在小程序 withCredentials 为 true 时或是在 App 调用 uni.getUserInfo，要求此前有调用过 uni.login 且登录态尚未过期。

**success 返回参数说明**

| **参数** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| userInfo | OBJECT | 用户信息对象 |  |
| rawData | String | 不包括敏感信息的原始数据字符串，用于计算签名。 |  |
| signature | String | 使用 sha1( rawData + sessionkey ) 得到字符串，用于校验用户信息。 | 微信小程序、头条小程序 |
| encryptedData | String | 包括敏感数据在内的完整用户信息的加密数据，详细见加密数据解密算法。 | 微信小程序、头条小程序 |
| iv | String | 加密算法的初始向量，详细见加密数据解密算法。 | 微信小程序、头条小程序 |
| errMsg | String | 描述信息 |  |

**userInfo 参数说明**

| **参数** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| nickName | String | 用户昵称 |  |
| openId | String | 该服务商唯一用户标识 | App |
| avatarUrl | String | 用户头像 |  |

除了以上三个必有的信息外，不同服务供应商返回的其它信息会存在差异。

#### [App端登陆的扩展说明](https://uniapp.dcloud.io/api/plugins/login?id=app%e7%ab%af%e7%99%bb%e9%99%86%e7%9a%84%e6%89%a9%e5%b1%95%e8%af%b4%e6%98%8e)

App端还支持更多登陆相关API，如logout，[详见](https://www.html5plus.org/doc/zh_cn/oauth.html" \t "_blank)

App端登陆相关的SDK需要在manifest中配置：

1. 打开 manifest.json -> App模块权限配置，勾选 OAuth(登陆鉴权)。
2. 打开 manifest.json -> App SDK配置，查看到登陆鉴权。在说明中有蓝色链接，其中包括向微信、QQ、微博等平台申请sdk的链接。
3. 向微信、QQ、微博等平台申请到sdk的信息后，回填到manifest里。
4. 这些配置需要打包生效，真机运行仍然是HBuilder基座的设置，可使用[自定义基座包](http://ask.dcloud.net.cn/article/12723)。离线打包请参考离线打包文档在原生工程中配置。
5. 配置并打包后，通过uni.getProvider可以得到配置的结果列表，注意这里返回的是manifest配置的，与手机端是否安装微信、QQ、微博无关。

如果手机端未安装QQ、微博，调用时会启动这些平台的wap页面登陆，如果已安装相应客户端，会启动它们的客户端登陆。

**示例**

uni.login({

provider: 'weixin',

success: function (loginRes) {

console.log(loginRes.authResult);

// 获取用户信息

uni.getUserInfo({

provider: 'weixin',

success: function (infoRes) {

console.log('用户昵称为：' + infoRes.userInfo.nickName);

}

});

}

});

#### [App端集成其他登陆SDK如支付宝、淘宝、facebook登陆的说明](https://uniapp.dcloud.io/api/plugins/login?id=app%e7%ab%af%e9%9b%86%e6%88%90%e5%85%b6%e4%bb%96%e7%99%bb%e9%99%86sdk%e5%a6%82%e6%94%af%e4%bb%98%e5%ae%9d%e3%80%81%e6%b7%98%e5%ae%9d%e3%80%81facebook%e7%99%bb%e9%99%86%e7%9a%84%e8%af%b4%e6%98%8e)

1. 使用原生插件方式，可以集成三方sdk，原生插件开发文档见<https://ask.dcloud.net.cn/article/35428>。开发之前可以先去[插件市场](https://ext.dcloud.net.cn/)看下有没有做好的。
2. 支付宝登陆、淘宝登陆在插件市场均有插件，[支付宝登陆](https://ext.dcloud.net.cn/search?q=%E6%94%AF%E4%BB%98%E5%AE%9D%E7%99%BB%E9%99%86)、[淘宝登陆](https://ext.dcloud.net.cn/search?q=%E7%99%BE%E5%B7%9D)。
3. 也可以内嵌web-view组件，使用web登陆模式集成这些三方登陆

## 分享

### [uni.share(OBJECT)](https://uniapp.dcloud.io/api/plugins/share?id=share)

分享到社交平台（微信、QQ、微博、短信、邮件等）

**平台差异说明**

| **平台** | **说明** |
| --- | --- |
| App | 使用 uni.share 进行分享，需要在 manifest.json 里配置各平台分享所必需的的字段，如appid等 |
| 小程序 | 不支持方法调用，只能用户主动点击触发分享，可使用 <button open-type="share"> 和 onShareAppMessage 进行自定义 |
| H5 | 如果是普通浏览器，浏览器自带分享按钮；如果是在微信内嵌浏览器中调用js-sdk，[参考](https://ask.dcloud.net.cn/article/35380" \t "_blank) |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| provider | String | 是 | 分享服务提供商，通过 [uni.getProvider](https://uniapp.dcloud.io/api/plugins/provider) 获取。 |
| type | Number | 否 | 分享类型。默认图文 0，更多值参考下面说明。 |
| title | String | 否 | 标题 |
| scene | String | provider 为 weixin 时必选 | 场景，可取值参考下面说明。 |
| summary | String | type 为 1 时必选 | 摘要 |
| href | String | type 为 0 时必选 | 跳转链接 |
| imageUrl | String | type 为 0、2、5 时必选 | 图片地址。type为0时，推荐使用小于20Kb的图片 |
| mediaUrl | String | type 为 3、4 时必选 | 音视频地址 |
| miniProgram | Object | type 为 5 时必选 | 分享小程序必要参数 |
| success | Function | 否 | 接口调用成功的回调 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**type 值说明**

| **值** | **说明** | **provider 支持度** |
| --- | --- | --- |
| 0 | 图文 | weixin、sinaweibo |
| 1 | 纯文字 |  |
| 2 | 纯图片 |  |
| 3 | 音乐 | weixin、qq |
| 4 | 视频 | weixin、sinaweibo |
| 5 | 小程序 | weixin |

**scene 值说明**

| **值** | **说明** |
| --- | --- |
| WXSceneSession | 分享到聊天界面 |
| WXSenceTimeline | 分享到朋友圈 |
| WXSceneFavorite | 分享到微信收藏 |

**miniProgram 值说明**

| **值** | **类型** | **说明** |
| --- | --- | --- |
| id | String | 微信小程序原始id |
| path | String | 点击链接进入的页面 |
| type | Number | 微信小程序版本类型，可取值： 0-正式版； 1-测试版； 2-体验版。 默认值为0。 |
| webUrl | String | 兼容低版本的网页链接 |

**注意事项：**

* App端可调用手机的系统分享，实现所有注册分享的应用的呼起，比如短信、邮件等，具体参考[plus.share.sendWithSystem的API文档](http://www.html5plus.org/doc/zh_cn/share.html#plus.share.sendWithSystem)
* 插件市场有一个封装好的分享菜单，直接弹出底部图标菜单，并且没有遮挡层级问题，推荐使用，<https://ext.dcloud.net.cn/plugin?id=69>
* uni.share API 仅用于 App 平台。小程序平台的分享，请参考 [小程序分享指引](https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/share.html)。
* 分享到 QQ 必须含有 href 链接
* 分享文字到 QQ 时，title 必选
* 新浪微博仅支持分享本地音视频
* 仅支持分享微信小程序到微信聊天界面，想进入朋友圈需改为分享图片方式，在图片中包含小程序码
* 在 iOS 端，若未安装微博客户端，会启用微博的网页分享，此时不能分享图片
* 分享新浪微博不会返回正确的成功回调
* 不能直接分享到QQ空间，可以分享到QQ，然后在QQ的界面里选择QQ空间。
* 分享微信朋友圈多图，微信官方已经禁掉这个功能。可以考虑把多张图用canvas合并成一张图分享出去。
* 从APP分享到微信时，无法判断用户点击取消分享，因为微信官方禁掉了分享成功的返回值。

#### [分享到微信聊天界面](https://uniapp.dcloud.io/api/plugins/share?id=%e5%88%86%e4%ba%ab%e5%88%b0%e5%be%ae%e4%bf%a1%e8%81%8a%e5%a4%a9%e7%95%8c%e9%9d%a2)

**分享文字**

uni.share({

provider: "weixin",

scene: "WXSceneSession",

type: 1,

summary: "我正在使用HBuilderX开发uni-app，赶紧跟我一起来体验！",

success: function (res) {

console.log("success:" + JSON.stringify(res));

},

fail: function (err) {

console.log("fail:" + JSON.stringify(err));

}

});

**分享图片**

uni.share({

provider: "weixin",

scene: "WXSceneSession",

type: 2,

imageUrl: "https://img-cdn-qiniu.dcloud.net.cn/uniapp/images/uni@2x.png",

success: function (res) {

console.log("success:" + JSON.stringify(res));

},

fail: function (err) {

console.log("fail:" + JSON.stringify(err));

}

});

**分享图文**

href、imageUrl 为必选参数，title/summary 二选一，最好将这四个参数都选上。

uni.share({

provider: "weixin",

scene: "WXSceneSession",

type: 0,

href: "http://uniapp.dcloud.io/",

title: "uni-app分享",

summary: "我正在使用HBuilderX开发uni-app，赶紧跟我一起来体验！",

imageUrl: "https://img-cdn-qiniu.dcloud.net.cn/uniapp/images/uni@2x.png",

success: function (res) {

console.log("success:" + JSON.stringify(res));

},

fail: function (err) {

console.log("fail:" + JSON.stringify(err));

}

});

#### [分享到微信朋友圈](https://uniapp.dcloud.io/api/plugins/share?id=%e5%88%86%e4%ba%ab%e5%88%b0%e5%be%ae%e4%bf%a1%e6%9c%8b%e5%8f%8b%e5%9c%88)

**分享文字**

uni.share({

provider: "weixin",

scene: "WXSenceTimeline",

type: 1,

summary: "我正在使用HBuilderX开发uni-app，赶紧跟我一起来体验！",

success: function (res) {

console.log("success:" + JSON.stringify(res));

},

fail: function (err) {

console.log("fail:" + JSON.stringify(err));

}

});

**分享图片**

uni.share({

provider: "weixin",

scene: "WXSenceTimeline",

type: 2,

imageUrl: "https://img-cdn-qiniu.dcloud.net.cn/uniapp/images/uni@2x.png",

success: function (res) {

console.log("success:" + JSON.stringify(res));

},

fail: function (err) {

console.log("fail:" + JSON.stringify(err));

}

});

**分享图文**

href、imageUrl 为必选参数，title、summary 至少有一项。

uni.share({

provider: "weixin",

scene: "WXSenceTimeline",

type: 0,

href: "http://uniapp.dcloud.io/",

title: "uni-app分享",

summary: "我正在使用HBuilderX开发uni-app，赶紧跟我一起来体验！",

imageUrl: "https://img-cdn-qiniu.dcloud.net.cn/uniapp/images/uni@2x.png",

success: function (res) {

console.log("success:" + JSON.stringify(res));

},

fail: function (err) {

console.log("fail:" + JSON.stringify(err));

}

});

**App分享为微信小程序**（App中分享一个内容到微信好友，对方微信中呈现的是一个小程序卡片）

uni.share({

provider: 'weixin',

scene: "WXSceneSession",

type: 5,

imageUrl: 'https://img-cdn-qiniu.dcloud.net.cn/uniapp/app/share-logo@3.png',

title: '欢迎体验uniapp',

miniProgram: {

id: 'gh\_abcdefg',

path: 'pages/index/index',

type: 0,

webUrl: 'http://uniapp.dcloud.io'

},

success: ret => {

console.log(JSON.stringify(ret));

}

});

### [onShareAppMessage(OBJECT)](https://uniapp.dcloud.io/api/plugins/share?id=onshareappmessage)

小程序中用户点击分享后，在 js 中定义 onShareAppMessage 处理函数（和 onLoad 等生命周期函数同级），设置该页面的分享信息。

* 用户点击分享按钮的时候会调用。这个分享按钮可能是小程序原生菜单自带的分享按钮，也可能是开发者在页面中放置的分享按钮（\）；
* 此事件需要 return 一个Object，用于自定义分享内容。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | √ | √ |

| **参数** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| from | String | 分享事件来源：button（页面内分享按钮）、menu（右上角分享按钮） |  |
| target | Object | 如果 from 值是 button，则 target 是触发这次分享事件的 button，否则为 undefined |  |
| webViewUrl | String | 页面中包含 <web-view> 组件时，返回当前 <web-view> 的url | 微信小程序、支付宝小程序 |

此事件需要 return 一个 Object，用于自定义分享内容，其内容如下：

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| title | String | 是 | 分享标题 |  |
| path | String | 是 | 页面 path ，必须是以 / 开头的完整路径。 |  |
| imageUrl | String | 否 | 分享图标，路径可以是本地文件路径、代码包文件路径或者网络图片路径 |  |
| content | String | 否 | 百度小程序表现为：分享内容；支付宝小程序表现为：吱口令文案 | 百度小程序、支付宝小程序 |
| desc | String | 否 | 自定义分享描述 | 支付宝小程序 |
| bgImgUrl | String | 否 | 自定义分享二维码的背景图，建议大小750\*950（网络图片路径） | 支付宝小程序 |
| success | Function | 否 | 接口调用成功的回调函数 | 支付宝小程序、百度小程序 |
| fail | Function | 否 | 接口调用失败的回调函数 | 支付宝小程序、百度小程序 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） | 百度小程序 |

**示例代码**

export default {

onShareAppMessage(res) {

if (res.from === 'button') {// 来自页面内分享按钮

console.log(res.target)

}

return {

title: '自定义分享标题',

path: '/pages/test/test?id=123'

}

}

}

### [uni.showShareMenu(OBJECT)](https://uniapp.dcloud.io/api/plugins/share?id=showsharemenu)

小程序的原生菜单中显示分享按钮

**1.6.0 新增**

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | √ |

| **属性** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| withShareTicket | Boolean | 否 | 是否使用带 shareTicket 的转发，默认为 flase。[详情](https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/share.html" \t "_blank) | 微信小程序 |
| title | String | 否 | 分享标题 | 百度小程序 |
| content | String | 否 | 分享内容 | 百度小程序 |
| imageUrl | String | 否 | 分享图标 | 百度小程序 |
| path | String | 否 | 页面 path ，必须是以 / 开头的完整路径。 | 百度小程序 |
| success | Function | 否 | 接口调用成功的回调函数 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

### [uni.hideShareMenu(OBJECT)](https://uniapp.dcloud.io/api/plugins/share?id=hidesharemenu)

小程序的原生菜单中隐藏分享按钮

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** |
| --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | x | √ |

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | function | 否 | 接口调用成功的回调函数 |
| fail | function | 否 | 接口调用失败的回调函数 |
| complete | function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**代码示例**

uni.hideShareMenu()

### [App 端各平台分享配置说明](https://uniapp.dcloud.io/api/plugins/share?id=app-%e7%ab%af%e5%90%84%e5%b9%b3%e5%8f%b0%e5%88%86%e4%ba%ab%e9%85%8d%e7%bd%ae%e8%af%b4%e6%98%8e)

* 第一步，打开 manifest.json -> App模块权限配置，勾选 Share(分享)；
* 第二步，按如下文档具体配置微信、微博、QQ的参数。

#### [微信分享](https://uniapp.dcloud.io/api/plugins/share?id=%e5%be%ae%e4%bf%a1%e5%88%86%e4%ba%ab)

在 manifest.json 的 App SDK 配置里，勾选微信消息及朋友圈，并填写 appid，如需在iOS平台使用还需要配置通用链接。

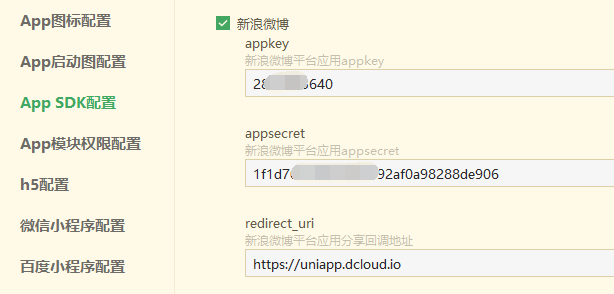
**参考文档**

* 微信 appid 申请步骤：<https://ask.dcloud.net.cn/article/208>。
* iOS平台微信SDK配置通用链接：<https://ask.dcloud.net.cn/article/36445>。



#### [新浪微博分享](https://uniapp.dcloud.io/api/plugins/share?id=%e6%96%b0%e6%b5%aa%e5%be%ae%e5%8d%9a%e5%88%86%e4%ba%ab)

在 manifest.json 的 App SDK 配置里，勾选勾选新浪微博，并填写相关appkey，新浪微博 appkey 申请步骤可参考：<https://ask.dcloud.net.cn/article/209>。



#### [QQ 分享](https://uniapp.dcloud.io/api/plugins/share?id=qq-%e5%88%86%e4%ba%ab)

在 manifest.json 的 App SDK 配置里，勾选分享到QQ好友，并填写相关appkey，QQ分享 appkey 申请步骤：

1. 前往 QQ 开放平台：<https://connect.qq.com/index.html>；
2. 完成开发者注册；
3. 创建应用，选择移动 App，填写相关信息，然后等待审核，审核通过后即可得到AppId。



这些配置需要打包生效，真机运行仍然是HBuilder基座的设置，可使用[自定义基座包](http://ask.dcloud.net.cn/article/12723)。离线打包请参考离线打包文档在原生工程中配置。

配置并打包后，通过uni.getProvider可以得到配置的结果列表，注意这里返回的是manifest配置的，与手机端是否安装微信、QQ、微博无关。

如果手机端未安装QQ、微博，调用时会启动这些平台的wap页面分享，如果已安装相应客户端，会启动它们的客户端分享。

##### [FAQ](https://uniapp.dcloud.io/api/plugins/share?id=faq)

* Q：App端如何集成其他分享SDK，如facebook分享、twitter分享
* A：插件市场已有相关插件，[详见](https://ext.dcloud.net.cn/search?q=%E5%88%86%E4%BA%AB)；也可以根据原生插件教程自行开发，原生插件开发文档见<https://ask.dcloud.net.cn/article/35428>

## 支付

### [uni.requestPayment(OBJECT)](https://uniapp.dcloud.io/api/plugins/payment?id=requestpayment)

支付

uni.requestPayment是一个统一各平台的客户端支付API，不管是在某家小程序还是在App中，客户端均使用本API调用。

本API运行在各端时，会自动转换为各端的原生支付调用API。

虽然客户端API统一了，但各平台的支付申请开通、配置回填、服务器开发，仍然需要看各个平台本身的支付文档。

比如微信有App支付、小程序支付、H5支付等不同的申请入口和使用流程，对应到uni-app，在App端要申请和使用微信的App支付，而小程序端则申请和使用微信的小程序支付。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| √ | [说明](https://uniapp.dcloud.io/api/plugins/payment?id=h5-payment) | √ | √ | √ | √ | √ |

**OBJECT 参数说明**

| **参数名** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| provider | String | 是 | 服务提供商，通过 [uni.getProvider](https://uniapp.dcloud.io/api/plugins/provider) 获取。 |  |
| orderInfo | String/Object | 是 | 订单数据，[注意事项](https://uniapp.dcloud.io/api/plugins/payment?id=orderinfo) | App、支付宝小程序、百度小程序、头条小程序 |
| timeStamp | String | 微信小程序必填 | 时间戳从1970年1月1日至今的秒数，即当前的时间。 | 微信小程序 |
| nonceStr | String | 微信小程序必填 | 随机字符串，长度为32个字符以下。 | 微信小程序 |
| package | String | 微信小程序必填 | 统一下单接口返回的 prepay\_id 参数值，提交格式如：prepay\_id=xx。 | 微信小程序 |
| signType | String | 微信小程序必填 | 签名算法，暂支持 MD5。 | 微信小程序 |
| paySign | String | 微信小程序必填 | 签名，具体签名方案参见 [微信小程序支付文档](https://pay.weixin.qq.com/wiki/doc/api/wxa/wxa_api.php?chapter=7_7&index=3) | 微信小程序 |
| bannedChannels | Array<String> | 否 | 需要隐藏的支付方式，详见 [百度小程序支付文档](https://smartprogram.baidu.com/docs/develop/api/open_payment/#requestPolymerPayment/) | 百度小程序 |
| service | Number | 头条小程序必填 | 固定值：1（拉起小程序收银台）开发者如果不希望使用头条小程序收银台，service设置为3/4时，可以直接拉起微信/支付宝进行支付：service=3： 微信API支付，不拉起小程序收银台；service=4： 支付宝API支付，不拉起小程序收银台。其中service=3、4，仅在1.35.0.1+基础库(头条743+)支持 | 头条小程序 |
| \_debug | Number | 否 | 仅限调试用，上线前去掉该参数。\_debug=1时，微信支付期间可以看到中间报错信息，方便调试 | 头条小程序 |
| getOrderStatus | Function | 头条小程序必填 | 商户前端实现的查询支付订单状态方法（该方法需要返回个Promise对象）。 service=3、4时不需要传。 | 头条小程序 |
| success | Function | 否 | 接口调用成功的回调 |  |
| fail | Function | 否 | 接口调用失败的回调函数 |  |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

#### [注意事项](https://uniapp.dcloud.io/api/plugins/payment?id=%e6%b3%a8%e6%84%8f%e4%ba%8b%e9%a1%b9)

* 头条小程序支付接口调整使用时请注意[发起头条支付](https://developer.toutiao.com/dev/cn/mini-app/develop/open-capacity/payment/pay)

#### [orderInfo 注意事项](https://uniapp.dcloud.io/api/plugins/payment?id=orderinfo)

1. 百度小程序的 orderInfo 为 Object 类型，详细的数据结构，参考：[百度收银台支付](https://smartprogram.baidu.com/docs/develop/api/open_payment/#requestPolymerPayment/)。
2. 支付宝小程序的 orderInfo(支付宝的规范为 tradeNO) 为 String 类型，表示支付宝交易号。
3. 头条小程序的 orderInfo 为 Object 类型，详见：[发起头条支付](https://developer.toutiao.com/dev/cn/mini-app/develop/open-capacity/payment/pay)
4. 由于头条新版支付接口要求版本较高，在不支持新版支付接口的情况下仍会对应旧版支付接口，此时 orderInfo 对应头条小程序 data， 详见：[头条支付旧版接口](https://developer.toutiao.com/dev/cn/mini-app/develop/open-capacity/payment/requestpayment-deprecated)。用户可以使用 tt.pay 判断是否支持新版接口。另外需要注意头条小程序在1.35.0+版本基础库支持了 canIUse ，在1.19.4+版本基础库支持了新版支付接口 tt.pay ，所以应避免使用 canIUse 判断是否为新版接口。
5. App端，支付宝支付和微信支付 orderInfo 均为 String 类型。
6. App端，苹果应用内支付 orderInfo 为Object 类型，{productid: 'productid'}。

#### [H5 平台](https://uniapp.dcloud.io/api/plugins/payment?id=h5-payment)

* 普通浏览器平台的支付，仍然是常规web做法。uni-app未封装。
* 在普通浏览器里也可以调起微信进行支付，这个在微信叫做H5支付，此功能未开放给普通开发者，需向微信单独申请，[详见](https://pay.weixin.qq.com/wiki/doc/api/H5.php?chapter=15_1)
* 微信内嵌浏览器运行H5版时，可通过js sdk实现微信支付，需要引入一个单独的js，[详见](https://ask.dcloud.net.cn/article/35380" \t "_blank)

**各平台支持的支付情况说明**

* 微信小程序里只支持微信小程序支付，在 [微信商户平台](https://pay.weixin.qq.com/) 申请支付时，选择公众号支付。
* App 里支持微信sdk支付、支付宝sdk支付、苹果iap应用内支付，在各平台申请支付时选择 App 支付。
* 其他支付（如银联）请使用web-view组件以H5方式实现。
* 支付宝小程序只支持支付宝支付。
* 百度小程序为百度支付，其二次封装了度小满、支付宝、微信支付。
* Hello uniapp 里演示了各种支付（iap支付见社区文章底部的赞助按钮），同时该示例开源了对应的服务端源码，参考：<https://github.com/dcloudio/H5P.Server/tree/master/payment>。

### [App平台支付流程](https://uniapp.dcloud.io/api/plugins/payment?id=app%e5%b9%b3%e5%8f%b0%e6%94%af%e4%bb%98%e6%b5%81%e7%a8%8b)

流程：支付平台功能申请 -> manifest.json 里配置支付参数 -> uni-app 里调用 API 进行支付

#### [App支付功能申请](https://uniapp.dcloud.io/api/plugins/payment?id=app%e6%94%af%e4%bb%98%e5%8a%9f%e8%83%bd%e7%94%b3%e8%af%b7)

1. 支付宝App支付功能申请

登录支付宝账号，创建应用接入支付宝App支付能力，包括以下步骤：

* + 创建应用（获取appid）
  + 开通App支付功能
  + 配置密钥（获取公钥、私钥）

具体可参考支付宝官方文档： [App支付快速接入](https://docs.open.alipay.com/204/105297/)

1. 微信App支付功能申请
   * 到 [微信开放平台](https://open.weixin.qq.com/) 申请移动应用并开通支付功能，申请应用后可以获取 AppID 和 AppSecret 值
   * 应用接入 [微信商户平台](https://pay.weixin.qq.com/)，选择 App 支付
   * 开通支付功能后可获取支付业务服务器配置数据：PARTNER（财付通商户号）、PARTNER\_KEY（财付通密钥）、PAYSIGNKEY（支付签名密钥）
   * 需要将从微信开放平台申请的appid，填回到 manifest-App SDK配置-支付-微信支付 中。打包后生效。

具体可参考微信官方文档： [移动应用开发](https://open.weixin.qq.com/cgi-bin/frame?t=home/app_tmpl&lang=zh_CN)

注意微信的App支付、小程序支付、H5支付是不同的体系。微信小程序支付在 [微信商户平台](https://pay.weixin.qq.com/) 申请支付时，选择公众号支付；普通浏览器里也可以调起微信进行支付，这个在微信叫做H5支付，此功能未开放给普通开发者，需向微信单独申请，[详见](https://pay.weixin.qq.com/wiki/doc/api/H5.php?chapter=15_1)

1. 苹果iap应用内支付申请

使用苹果开发者账号登录 [App Store Connect](https://appstoreconnect.apple.com/)，在应用的功能选项卡页面，添加 App 内购项目。注意：

* + 内购项目的各信息需要填写完整，然后保存，此时内购项目的状态应该是准备提交，当提交应用通过审核后，状态则变为已批准
  + 测试时，建议使用测试证书打一个自定义的 iOS 基座进行测试
  + 在应用 TestFight 的选项卡添加 App Store Connect 用户，测试支付时可以使用此用户帐号进行测试
  + orderInfo 的 productid 是自己填写的产品 ID
  + 调用 uni.requestPayment 前必须先使用 5+Plus 的方法调用 requestOrder 获取订单信息，否则会导致无法支付
  + 更多可参考：<https://ask.dcloud.net.cn/article/497>

#### [manifest.json里配置相关参数](https://uniapp.dcloud.io/api/plugins/payment?id=manifestjson%e9%87%8c%e9%85%8d%e7%bd%ae%e7%9b%b8%e5%85%b3%e5%8f%82%e6%95%b0)

1. 在manifest.json - App模块权限选择 中勾选 payment(支付)
2. 在 manifest.json - App SDK配置 中，勾选需要的支付平台，目前有微信支付、支付宝支付、苹果应用内支付(IAP)，其中微信支付需要填写从微信开放平台获取的AppID 
3. 这些配置需要打包生效，真机运行仍然是HBuilder基座的设置，可使用自定义基座调试。离线打包请参考离线打包文档在原生工程中配置。
4. 配置并打包后，通过uni.getProvider可以得到配置的结果列表，注意这里返回的是manifest配置的，与手机端是否安装微信、支付宝无关。

如果手机端未安装支付宝，调用时会启动支付宝的wap页面登陆，如果已安装相应客户端，会启动其客户端登陆。

#### [uni-app里开发](https://uniapp.dcloud.io/api/plugins/payment?id=uni-app%e9%87%8c%e5%bc%80%e5%8f%91)

* 前端：使用 uni.request 请求服务端接口，得到订单数据，使用 uni.requestPayment 向支付平台发起支付请求，拉起支付平台的客户端进行支付。在hello uni-app里详细代码。
* 服务端：PHP可参考 <https://github.com/dcloudio/H5P.Server/tree/master/payment>。

#### [FAQ](https://uniapp.dcloud.io/api/plugins/payment?id=faq)

* Q：如何使用ping++等聚合支付  
  A：uni-app的js API 已经完成跨端统一，客户端无需使用三方聚合支付。仅在服务器端使用三方聚合支付即可。
* Q：App端如何使用其他支付，比如银联、PayPal。  
  A：1、可以在web-view组件里使用它们的wap版支付；2、可以集成原生sdk，插件市场均有，[详见](https://ext.dcloud.net.cn/search?q=%E6%94%AF%E4%BB%98)。也可以自行开发原生插件，开发文档见<https://ask.dcloud.net.cn/article/35428>。
* Q：Appstore审核报PGPay SDK不允许上架的问题  
  A：数字类产品（比如购买会员等不需要配送实物的商品），Apple规定必须使用苹果IAP应用内支付，给Apple分成30%。打包的时候不要勾选微信或支付宝等其他支付方式。如果你提交的包里包含了微信支付宝等支付的sdk，即使没使用，Appstore也会认为你有隐藏方式，以后会绕过iap，不给Apple分成，因此拒绝你的App上线。云打包时，manifest里选上支付模块，但sdk配置里去掉微信支付和支付宝支付。很多开发者的Android版是包含微信和支付宝支付的，此时注意分开判断。详见<https://ask.dcloud.net.cn/article/36447>

**示例**

App 支付

uni.requestPayment({

provider: 'alipay',

orderInfo: 'orderInfo', //微信、支付宝订单数据

success: function (res) {

console.log('success:' + JSON.stringify(res));

},

fail: function (err) {

console.log('fail:' + JSON.stringify(err));

}

});

微信小程序支付

// 仅作为示例，非真实参数信息。

uni.requestPayment({

provider: 'wxpay',

timeStamp: String(Date.now()),

nonceStr: 'A1B2C3D4E5',

package: 'prepay\_id=wx20180101abcdefg',

signType: 'MD5',

paySign: '',

success: function (res) {

console.log('success:' + JSON.stringify(res));

},

fail: function (err) {

console.log('fail:' + JSON.stringify(err));

}

});

苹果应用内支付

<template>

<view>

<view class="uni-list">

<radio-group @change="applePriceChange">

<label class="uni-list-cell uni-list-cell-pd" v-for="(item, index) in priceList" :key="index">

{{item.text}}

<radio :value="item.value" :checked="item.checked" />

</label>

</radio-group>

</view>

<view class="uni-padding-wrap">

<button class="ipaPayBtn" @click="requestPayment" :loading="loading" :disabled="disabled">确认支付</button>

</view>

</view>

</view>

</template>

<script>

let iapChannel = null,

productId = 'HelloUniappPayment1',

productIds = ['HelloUniappPayment1', 'HelloUniappPayment6'];

export default {

data() {

return {

title: 'request-payment',

loading: false,

disabled: true,

priceList: [{

value: 'HelloUniappPayment1',

text: '支付1元',

checked: true

}, {

value: 'HelloUniappPayment6',

text: '支付6元',

checked: false

}]

}

},

onLoad: function() {

plus.payment.getChannels((channels) => {

console.log("获取到channel" + JSON.stringify(channels))

for (var i in channels) {

var channel = channels[i];

if (channel.id === 'appleiap') {

iapChannel = channel;

this.requestOrder();

}

}

if(!iapChannel){

this.errorMsg()

}

}, (error) => {

this.errorMsg()

});

},

methods: {

requestOrder() {

uni.showLoading({

title:'检测支付环境...'

})

iapChannel.requestOrder(productIds, (orderList) => { //必须调用此方法才能进行 iap 支付

this.disabled = false;

console.log('requestOrder success666: ' + JSON.stringify(orderList));

uni.hideLoading();

}, (e) => {

console.log('requestOrder failed: ' + JSON.stringify(e));

uni.hideLoading();

this.errorMsg()

});

},

requestPayment(e) {

this.loading = true;

uni.requestPayment({

provider: 'appleiap',

orderInfo: {

productid: productId

},

success: (e) => {

uni.showModal({

content: "感谢您的赞助",

showCancel: false

})

},

fail: (e) => {

uni.showModal({

content: "支付失败,原因为: " + e.errMsg,

showCancel: false

})

},

complete: () => {

console.log("payment结束")

this.loading = false;

}

})

},

applePriceChange(e) {

productId = e.detail.value;

},

errorMsg(){

uni.showModal({

content: "暂不支持苹果 iap 支付",

showCancel: false

})

}

}

}

</script>

## 推送

push 是指从服务器主动给手机端发送消息。

* App平台

uni-app 提供了 uni push 服务，这是一个包括客户端和服务器的统一服务，整合了苹果APNs、华为、小米、OPPO、VIVO、魅族等多家厂商的推送和个推的独立推送，不管客户端还是服务器，一套代码多端推送。

开发者可以在uniPush中只接入个推，不配置其他厂商服务，但这将导致Android上App离线后无法推送消息。因为现在的国产rom节电管理策略严格，不使用rom厂商的push基本难以离线送达消息。 若想提供消息推送成功率，只有使用uniPush，并向各rom厂商申请开通推送服务（免费）。

因本文档内容有限，另行开贴，介绍业务、开通流程，请务必仔细阅读 <https://ask.dcloud.net.cn/article/35622>。

* 客户端调用的js API见：[https://www.html5plus.org/doc/zh\_cn/push.html](https://www.html5plus.org/doc/zh_cn/push.html" \t "_blank)
* 服务器调用接口文档，仍然是个推的服务器文档[http://docs.getui.com/](http://docs.getui.com/" \t "_blank)
* web发送界面，在DCloud的开发者后台：[https://dev.dcloud.net.cn/](https://dev.dcloud.net.cn/" \t "_blank)

### 其他相关资源

* 检查应用是否被授予推送权限：<https://ext.dcloud.net.cn/plugin?id=594>
* 开启关闭推送服务：<https://ext.dcloud.net.cn/plugin?id=727>
* 自定义iOS推送铃声：<https://ext.dcloud.net.cn/plugin?id=690>
* 如何自定义推送通知的图标：<https://ask.dcloud.net.cn/article/35537>

插件市场也提供了其他三方推送方案，但注意unipush是推送成功率更高的解决方案，并且免费使用。不推荐开发者使用其他推送服务。

但我们发现很多开发者有误解，导致还在错误使用其他推送。

* 常见误解1：“uniPush的专业性，和专业的个推、极光等服务可相比吗？” 答：uniPush是由个推将其本来收费的vip push产品，免费提供给了DCloud的开发者。它与个推vip push的只有2个区别：1、免费；2、账户使用的是DCloud开发者账户，而无需再重新注册个推账户。个推是A股上市公司，专业性在推送领域领先。
* 常见误解2：“uniPush好麻烦，我就喜欢个推、极光这种简单sdk，不想去各个rom厂商去申请一圈” 答：有此误解的开发者，根本不知道推送行业的现状。不集成rom厂商的推送，就无法在App离线时发送push。开发者可以只使用uniPush里面的个推服务，不去各个rom厂商申请推送，但你要知道这样做的后果就是在华为、小米、OPPO、VIVO、魅族上发不了离线消息。
* 常见误解3：“uniPush的送达率还是不够，是否可以付费来提升送达率，个推是有付费提升送达率的方法的” 答：前文已经说了。个推的付费提升送达率的产品就是vip push，而uniPush就是个推的vip Push。DCloud通过谈判免费给DCloud的开发者使用了。
* 小程序平台

小程序平台的类似概念叫做模板消息。

以微信为例，开发者的服务器发送消息给微信的服务器，微信服务器会发送一条模板消息，折叠到微信的消息列表中的服务通知里。它属于后台开发，和手机端无关。

微信模板消息文档：<https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/template-message.html>

支付宝模板消息文档：<https://docs.alipay.com/mini/introduce/message>

百度模板消息文档：<https://smartprogram.baidu.com/docs/develop/third/api/>

##### [FAQ](https://uniapp.dcloud.io/api/plugins/push?id=faq)

Q：为什么真机测试推送可以用，打包后就失败呢？  
A：HBuilder 真机运行环境下，第三方支付 SDK 的配置信息是 HBuilder 这个应用。这个配置信息是不能动态修改的，因此涉及到第三方 SDK 的配置，需要打包自定义基座进行测试。[真机运行自定义基座包使用说明](http://ask.dcloud.net.cn/article/12723)

Q：用到了推送功能，在打包原生应用时，需要注意什么呢？  
A：如果是云打包，首先勾选权限配置，manifest.json->App 模块权限配置->Push。然后，manifest.json->App SDK 配置->推送，按照提示申请相应平台的信息，并填写。最后，勾选并填写完成后再进行打包即可。如果是离线打包，自行在原生工程中配置SDK，并确保选择了推送模块。

## 语音

#### [voice](https://uniapp.dcloud.io/api/plugins/voice?id=voice)

voice 包括语言识别和语音朗读两部分。

仅百度小程序平台、App平台支持，各平台开发方式暂未统一，使用时需注意用[条件编译](https://uniapp.dcloud.io/platform)调用不同平台的代码。

百度小程序平台支持语音识别，参考：[规范详情](https://smartprogram.baidu.com/docs/develop/api/ai_voice/)

App 平台实现参考：

* 语音识别：支持科大讯飞语音识别和百度语音识别，云打包的话需要在manifest中勾选模块和App SDK配置，开发规范见：<https://www.html5plus.org/doc/zh_cn/speech.html>，配置文档及讯飞百度的差别见：<https://ask.dcloud.net.cn/article/35059>
* 语音朗读：调用科大讯飞进行语音合成、tts朗读，参考<https://ask.dcloud.net.cn/article/1081>

# 平台扩展

## 原生APP插件

### [uni.requireNativePlugin(PluginName)](https://uniapp.dcloud.io/api/extend/native-plugin?id=requirenativeplugin)

引入 App 原生插件。

平台差异说明：

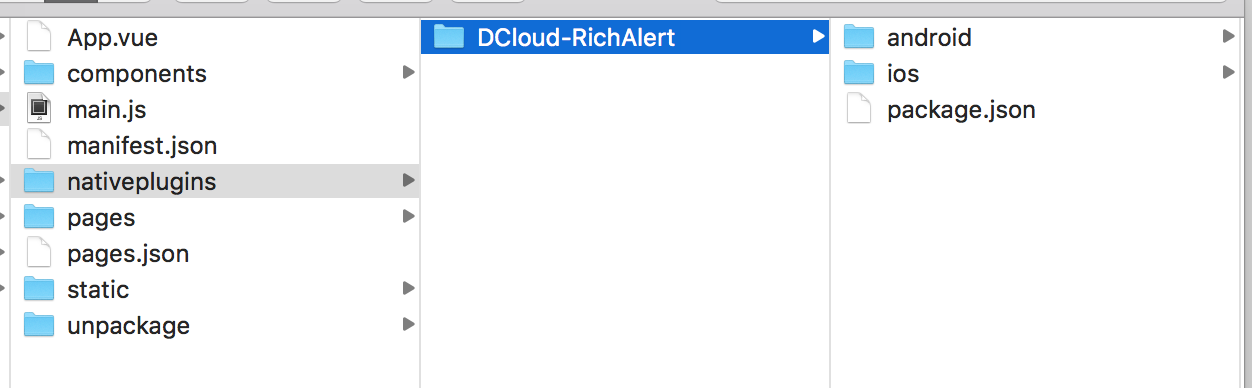
* App

自 HXuilderX1.4 版本起，uni-app 支持引入原生插件，使用方式如下：

const PluginName = uni.requireNativePlugin(PluginName); // PluginName 为原生插件名称

不管是vue文件还是nvue文件，都是这个API。

下面以原生插件 [DCloud-RichAlert](https://ext.dcloud.net.cn/plugin?id=36" \t "_blank) 为例，介绍如何使用此API。

1. 在 uni-app 项目工程中添加插件：从 <http://ext.dcloud.net.cn/plugin?id=36> 下载 DCloud-RichAlert.zip ，解压到 uni-app 项目根目录下的 nativeplugins 目录（如不存在则创建），添加后目录结构为：
2. 在页面引入原生插件，uni.requireNativePlugin 使用后返回一个对象：

const dcRichAlert = uni.requireNativePlugin('DCloud-RichAlert')

1. 使用原生插件
2. dcRichAlert.show({
3. position: 'bottom',
4. title: "提示信息",
5. titleColor: '#FF0000',
6. content: "<a href='https://uniapp.dcloud.io/' value='Hello uni-app'>uni-app</a> 是一个使用 Vue.js 开发跨平台应用的前端框架!\n免费的\n免费的\n免费的\n重要的事情说三遍",
7. contentAlign: 'left',
8. checkBox: {
9. title: '不再提示',
10. isSelected: true
11. },
12. buttons: [{
13. title: '取消'
14. }, {
15. title: '否'
16. }, {
17. title: '确认',
18. titleColor: '#3F51B5'
19. }]
20. }, result => {
21. console.log(result)

});

**注意事项：**

1. 插件来源为 [插件市场](http://ext.dcloud.net.cn/?cat1=5)，或自己开发的插件，Android插件开发参考：<https://ask.dcloud.net.cn/article/35416>，iOS插件开发参考：<https://ask.dcloud.net.cn/article/35415>；
2. 不支持真机运行原生插件，仅支持云端打包运行或使用自定义基座运行；
3. 如果插件需要传递文件路径，则需要传手机文件的绝对路径，可使用 5+ [IO模块](http://www.html5plus.org/doc/zh_cn/io.html) 的相关 API 得到文件的绝对路径。

# 其他

## 授权

### [uni.authorize(OBJECT)](https://uniapp.dcloud.io/api/other/authorize?id=authorize)

提前向用户发起授权请求。调用后会立刻弹窗询问用户是否同意授权小程序使用某项功能或获取用户的某些数据，但不会实际调用对应接口。如果用户之前已经同意授权，则不会出现弹窗，直接返回成功。如果用户之前拒绝了授权，此接口会直接进入失败回调，一般搭配uni.getSetting和uni.openSetting使用。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | √ | √ |

注意：App平台的授权判断方式，另见：<https://ext.dcloud.net.cn/plugin?id=594>

**OBJECT 参数说明**

| **参数** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| scope | String | 是 | 需要获取权限的 scope，详见 scope 列表。 |
| success | Function | 否 | 接口调用成功的回调函数 |
| fail | Function | 否 | 接口调用失败的回调函数 |
| complete | Function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

##### [scope 列表](https://uniapp.dcloud.io/api/other/authorize?id=scope-%e5%88%97%e8%a1%a8)

| **scope** | **对应接口** | **描述** | **平台差异说明** |
| --- | --- | --- | --- |
| scope.userInfo | [uni.getUserInfo](https://uniapp.dcloud.io/api/plugins/login?id=getuserinfo) | 用户信息 |  |
| scope.userLocation | [uni.getLocation](https://uniapp.dcloud.io/api/location/location?id=getlocation), [uni.chooseLocation](https://uniapp.dcloud.io/api/location/location?id=chooselocation) | 地理位置 |  |
| scope.userLocationBackground | wx.userLocationBackground | 后台定位 | 微信小程序 |
| scope.address | [uni.chooseAddress](https://uniapp.dcloud.io/api/other/choose-address) | 通信地址 |  |
| scope.record | [uni.getRecorderManager](https://uniapp.dcloud.io/api/media/record-manager?id=getrecordermanager) | 录音功能 |  |
| scope.writePhotosAlbum | [uni.saveImageToPhotosAlbum](https://uniapp.dcloud.io/api/media/image?id=saveimagetophotosalbum), [uni.saveVideoToPhotosAlbum](https://uniapp.dcloud.io/api/media/video?id=savevideotophotosalbum) | 保存到相册 | 头条小程序的返回值是scope.album |
| scope.camera | [<camera />](https://uniapp.dcloud.io/component/camera) 组件，头条下的扫码、拍照、选择相册 | 摄像头 |  |
| scope.invoice | [wx.chooseInvoice](https://developers.weixin.qq.com/miniprogram/dev/api/wx.chooseInvoice.html) | 获取发票 | 微信小程序、QQ小程序 |
| scope.invoiceTitle | [uni.chooseInvoiceTitle](https://uniapp.dcloud.io/api/other/invoice-title) | 发票抬头 | 微信小程序、百度小程序、QQ小程序 |
| scope.werun | [wx.getWeRunData](https://developers.weixin.qq.com/miniprogram/dev/api/wx.getWeRunData.html) | 微信运动步数 | 微信小程序 |

**代码示例**

uni.authorize({

scope: 'scope.userLocation',

success() {

uni.getLocation()

}

})

## 设置

### [uni.openSetting(OBJECT)](https://uniapp.dcloud.io/api/other/setting?id=opensetting)

调起客户端小程序设置界面，返回用户设置的操作结果。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | √ | √ | √ |

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | function | 否 | 接口调用成功的回调函数 |
| fail | function | 否 | 接口调用失败的回调函数 |
| complete | function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| authSetting | Object | 用户授权结果，其中 key 为 [scope](https://uniapp.dcloud.io/api/other/authorize?id=scope-%e5%88%97%e8%a1%a8) 值，value 为 Boolean 值，表示用户是否允许授权 |

**代码示例**

uni.openSetting({

success(res) {

console.log(res.authSetting)

}

});

### [uni.getSetting(OBJECT)](https://uniapp.dcloud.io/api/other/setting?id=getsetting)

获取用户的当前设置。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | √ | √ | √ |

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | function | 否 | 接口调用成功的回调函数 |
| fail | function | 否 | 接口调用失败的回调函数 |
| complete | function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 返回参数**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| authSetting | Object | 用户授权结果，其中 key 为 [scope](https://uniapp.dcloud.io/api/other/authorize?id=scope-%e5%88%97%e8%a1%a8) 值，value 为 Boolean 值，表示用户是否允许授权 |

#### [示例代码](https://uniapp.dcloud.io/api/other/setting?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81)

uni.getSetting({

success(res) {

console.log(res.authSetting)

}

})

## 收货地址

### [uni.chooseAddress(OBJECT)](https://uniapp.dcloud.io/api/other/choose-address?id=chooseaddress)

获取用户收货地址。调起用户编辑收货地址原生界面，并在编辑完成后返回用户选择的地址，需要用户授权 scope.address。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | √ | √ | x |

**OBJECT参数说明**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | function | 否 | 接口调用成功的回调函数 |
| fail | function | 否 | 接口调用失败的回调函数 |
| complete | function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success返回参数说明**

| **属性** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| userName | string | 收货人姓名 |  |
| postalCode | string | 邮编 |  |
| provinceName | string | 国标收货地址第一级地址 |  |
| cityName | string | 国标收货地址第一级地址 |  |
| detailInfo | string | 详细收货地址信息 |  |
| nationalCode | string | 收货地址国家码 |  |
| telNumber | string | 收货人手机号码 |  |
| errMsg | string | 错误信息 | 微信小程序 |

**示例代码**

uni.chooseAddress({

success(res) {

console.log(res.userName)

console.log(res.postalCode)

console.log(res.provinceName)

console.log(res.cityName)

console.log(res.countyName)

console.log(res.detailInfo)

console.log(res.nationalCode)

console.log(res.telNumber)

}

})

## 获取发票抬头

### [uni.chooseInvoiceTitle(OBJECT)](https://uniapp.dcloud.io/api/other/invoice-title?id=chooseinvoicetitle)

选择用户的发票抬头，需要用户授权 scope.invoiceTitle。

在微信小程序中，当前当前小程序必须关联一个公众号，且这个公众号是完成了[微信认证](https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1496554031_RD4xe)的，才能调用 chooseInvoiceTitle。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | x |

**OBJECT参数说明**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| success | function | 否 | 接口调用成功的回调函数 |
| fail | function | 否 | 接口调用失败的回调函数 |
| complete | function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success返回参数说明**

| **属性** | **类型** | **说明** | **平台差异说明** |
| --- | --- | --- | --- |
| type | string | 抬头类型（0：单位，1：个人） |  |
| title | string | 抬头名称 |  |
| taxNumber | string | 抬头税号 |  |
| companyAddress | string | 单位地址 |  |
| telephone | string | 手机号码 |  |
| bankName | string | 银行名称 |  |
| bankAccount | string | 银行账号 |  |
| errMsg | string | 错误信息 | 微信小程序 |

**示例代码**

uni.chooseInvoiceTitle({

success(res) {

console.log(res.type);

console.log(res.title);

console.log(res.taxNumber);

console.log(res.companyAddress);

console.log(res.telephone);

console.log(res.bankName);

console.log(res.bankAccount);

}

})

## 小程序跳转

### [uni.navigateToMiniProgram(OBJECT)](https://uniapp.dcloud.io/api/other/open-miniprogram?id=navigatetominiprogram)

打开另一个小程序。

**平台差异说明**

| **App** | **H5** | [微信小程序](https://developers.weixin.qq.com/miniprogram/dev/api/open-api/miniprogram-navigate/wx.navigateToMiniProgram.html) | [支付宝小程序](https://docs.alipay.com/mini/api/open-miniprogram) | [百度小程序](https://smartprogram.baidu.com/docs/develop/api/open_smartprogram/#swan-navigateToSmartProgram/) | [头条小程序](https://developer.toutiao.com/dev/cn/mini-app/develop/open-capacity/mini-app-forward/navigatetominiprogram) | [QQ小程序](https://q.qq.com/wiki/develop/miniprogram/API/open_port/port_change.html#qq-navigatetominiprogram) |
| --- | --- | --- | --- | --- | --- | --- |
| x(见下) | x | √ | √ | √ | √(1.15.0+) | √ |

* App平台打开微信小程序，使用plus.share的[launchMiniProgram](https://www.html5plus.org/doc/zh_cn/share.html" \l "plus.share.ShareService.launchMiniProgram" \t "_blank)。注意uni-app不需要plus ready，将plus ready里的代码写到页面的onLoad生命周期即可。使用此功能需在manifest中配置微信分享SDK信息，打包后生效。
* 各小程序平台对跳转到其他小程序有一些限制和规定，需要遵守，具体见各平台文档。

**OBJECT 参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- | --- |
| appId | string |  | 是 | 要打开的小程序 appId（百度小程序则填写App Key） |  |
| path | string |  | 否 | 打开的页面路径，如果为空则打开首页 |  |
| extraData | object |  | 否 | 需要传递给目标小程序的数据，目标小程序可在 App.vue 的 onLaunch或onShow 中获取到这份数据。 |  |
| envVersion | string | release | 否 | 要打开的小程序版本，有效值： develop（开发版），trial（体验版），release（正式版）。仅在当前小程序为开发版或体验版时此参数有效。如果当前小程序是正式版，则打开的小程序必定是正式版。 | 支付宝小程序、微信小程序 |
| success | function |  | 否 | 接口调用成功的回调函数 |  |
| fail | function |  | 否 | 接口调用失败的回调函数 |  |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |  |

**示例代码**

uni.navigateToMiniProgram({

appId: '',

path: 'pages/index/index?id=123',

extraData: {

'data1': 'test'

},

success(res) {

// 打开成功

}

})

### [uni.navigateBackMiniProgram(OBJECT)](https://uniapp.dcloud.io/api/other/open-miniprogram?id=navigatebackminiprogram)

跳转回上一个小程序，只有当另一个小程序跳转到当前小程序时才会能调用成功。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | √ | √ | √ |

**OBJECT参数说明**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| extraData | Object | 否 | 需要返回给上一个小程序的数据，上一个小程序可在 App.vue 的 onShow 中获取到这份数据 |
| success | function | 否 | 接口调用成功的回调函数 |
| fail | function | 否 | 接口调用失败的回调函数 |
| complete | function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**示例代码**

uni.navigateBackMiniProgram({

extraData: {

'data1': 'test'

},

success(res) {

// 返回成功

}

})

## 账号信息

#### [uni.getAccountInfoSync()](https://uniapp.dcloud.io/api/other/getAccountInfoSync?id=getaccountinfosync)

获取当前帐号信息，可以返回小程序的Appid。如果使用了微信小程序的云端插件，还可以反馈插件的id和版本

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | √ |

**返回值**

**Object**

帐号信息

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| miniProgram | Object | 小程序帐号信息 |
| plugin | Object | 插件帐号信息（仅在插件中调用时包含这一项） |
| appName | string | 智能小程序名称 (仅百度小程序支持) |
| lastAppURL | string | 智能小程序最近一次打开的调起协议 (仅百度小程序支持) |
| scheme | string | 调起协议的协议头 (仅百度小程序支持) |

**miniProgram 的结构**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| appId | string | 小程序 appId |

**plugin 的结构**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| appId | string | 插件 appId (百度小程序不支持) |
| version | string | 插件版本号 |

**示例代码**

const accountInfo = uni.getAccountInfoSync();

console.log(accountInfo.miniProgram.appId); // 小程序 appId

console.log(accountInfo.plugin.appId); // 插件 appId

console.log(accountInfo.plugin.version); // 插件版本号， 'a.b.c' 这样的形式

## 运动(计步器)

#### [sport 运动](https://uniapp.dcloud.io/api/other/sport?id=sport-%e8%bf%90%e5%8a%a8)

此功能为计步器，用于获取手机用户的运动步数。

各平台开发方式暂未统一，使用时需注意用[条件编译](https://uniapp.dcloud.io/platform)调用不同平台的代码。

* App 平台：需使用原生插件，详见[计步器插件](https://ext.dcloud.net.cn/search?q=%E8%AE%A1%E6%AD%A5%E5%99%A8)
* 微信小程序平台：[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/open-api/werun/wx.getWeRunData.html)
* 支付宝小程序平台：[规范详情](https://docs.alipay.com/mini/api/gxuu7v)

## 统计

#### [统计](https://uniapp.dcloud.io/api/other/report?id=%e7%bb%9f%e8%ae%a1)

从uni-app 2.2.3起，支持uni统计。一张报表，掌握全端数据。业务介绍详见[https://tongji.dcloud.net.cn](https://tongji.dcloud.net.cn/)

自定义事件是统计中不可获取的功能。开发者可通过本API自定义上报统计数据，如统计登录、注册、点击某个按钮，我们都可以称之为自定义事件。

##### [uni.report( eventName , options)](https://uniapp.dcloud.io/api/other/report?id=report)

**参数说明**

| **参数** | **类型** | **描述** |
| --- | --- | --- |
| eventName | String | 事件名称，最大长度不超过 255 个字符 |
| options | String 、 Object | 事件参数 |

Tips

* eventName 为 String 类型，并且字符长度必须小于255
* options 为 String 类型时，字符长度必须小于255
* options 为 Object 类型时，该对象的值只能为 String 类型
* 字符串支持特殊字符但不包括（英文逗号 , 英文冒号 : 点 .）
* eventName 为 title 时为内容标题上报，用户不能自定义。此时数据会展现在uni统计的首页-内容统计及左侧导航的内容统计中。方便查看内容页数据。
* 用户在使用 uni.login() 会执行登录事件，不携带参数。如果如需上报携带具体参数的数据，需要手动调用 uni.report('login',{...})
* 用户在使用 uni.share() 或触发 onShareAppMessage 会执行分享事件，不携带参数。如果如需上报携带具体参数的数据，需要手动调用 uni.report('share',{...})
* 用户在使用 uni.requestPayment() 会执行支付事件，不携带参数。如果如需上报携带具体参数的数据，需要手动调用 uni.report('pay\_success',{...}) 和 uni.report('pay\_fail',{...})

**示例**

// 内容统计

// 当 eventName 为 title 时，options 只能为 String 类型

uni.report('title','首页')

// 登录

uni.report('login',{

'name':'uni-app',

'age':'21',

// ...

})

// 分享

uni.report('share','分享')

// 支付成功

uni.report('pay\_success','支付成功')

// or

uni.report('pay\_success',{

"订单金额":'20元',

"订单名称":'鼠标',

// ...

})

// 支付失败

uni.report('pay\_fail','支付失败')

// or

uni.report('pay\_fail',{

"订单金额":'20元',

"订单名称":'鼠标',

// ...

})

// 注册

uni.report('register',{

'name':'uni-app',

'age':'21',

// ...

})

// 搜索

uni.report('search','搜索内容')

// or

uni.report('search',{

'内容':'搜索内容'

})

自定义事件上报后，在统计后台的事件和转换栏目中，可以看到上报的事件情况。

**TIPS:**

* 小程序平台，需将tongji.dcloud.net.cn配入域名白名单，[详见](https://ask.dcloud.net.cn/article/36298)
* uni统计使用教程：<https://ask.dcloud.net.cn/article/36303>

除了官方的uni统计，如果开发者还需要调用小程序平台的自定义统计，给小程序的后台上报数据，那么需使用条件编译，在各端调用此自己的API。

##### [微信小程序平台：](https://uniapp.dcloud.io/api/other/report?id=%e5%be%ae%e4%bf%a1%e5%b0%8f%e7%a8%8b%e5%ba%8f%e5%b9%b3%e5%8f%b0%ef%bc%9a)

* 数据上报：[wx.reportMonitor](https://developers.weixin.qq.com/miniprogram/dev/api/open-api/report/wx.reportMonitor.html)
* 数据分析：[wx.reportAnalytics](https://developers.weixin.qq.com/miniprogram/dev/api/open-api/data-analysis/wx.reportAnalytics.html)

##### [支付宝小程序平台：](https://uniapp.dcloud.io/api/other/report?id=%e6%94%af%e4%bb%98%e5%ae%9d%e5%b0%8f%e7%a8%8b%e5%ba%8f%e5%b9%b3%e5%8f%b0%ef%bc%9a)

* 自定义分析数据的上报接口：[my.reportAnalytics](https://docs.alipay.com/mini/api/report)

##### [百度小程序平台：](https://uniapp.dcloud.io/api/other/report?id=%e7%99%be%e5%ba%a6%e5%b0%8f%e7%a8%8b%e5%ba%8f%e5%b9%b3%e5%8f%b0%ef%bc%9a)

* 数据分析：[swan.reportAnalytics](https://smartprogram.baidu.com/docs/develop/api/data/#swan-reportAnalytics/)

##### [头条小程序平台：](https://uniapp.dcloud.io/api/other/report?id=%e5%a4%b4%e6%9d%a1%e5%b0%8f%e7%a8%8b%e5%ba%8f%e5%b9%b3%e5%8f%b0%ef%bc%9a)

* [reportAnalytics](https://developer.toutiao.com/dev/cn/mini-app/develop/open-capacity/data-analysis/reportanalytics)

##### [QQ小程序平台：](https://uniapp.dcloud.io/api/other/report?id=qq%e5%b0%8f%e7%a8%8b%e5%ba%8f%e5%b9%b3%e5%8f%b0%ef%bc%9a)

* 数据上报：[qq.reportMonitor](https://q.qq.com/wiki/develop/miniprogram/API/open_port/port_dataup.html)
* 数据分析：[qq.reportAnalytics](https://q.qq.com/wiki/develop/miniprogram/API/open_port/port_dataanalysis.html#qq-reportanalytics)

##### [App平台的友盟统计：](https://uniapp.dcloud.io/api/other/report?id=app%e5%b9%b3%e5%8f%b0%e7%9a%84%e5%8f%8b%e7%9b%9f%e7%bb%9f%e8%ae%a1%ef%bc%9a)

* 友盟统计：[开发规范](http://www.html5plus.org/doc/zh_cn/statistic.html)，[配置文档](https://ask.dcloud.net.cn/article/74)

## 卡卷

#### [卡券](https://uniapp.dcloud.io/api/other/card?id=%e5%8d%a1%e5%88%b8)

仅微信小程序、支付宝小程序支持，各平台开发方式暂未统一，使用时需注意用[条件编译](https://uniapp.dcloud.io/platform)调用不同平台的代码。

* 微信小程序：[规范详情](https://developers.weixin.qq.com/miniprogram/dev/api/wx.openCard.html)
* 支付宝小程序：[规范详情](https://docs.alipay.com/mini/api/card-voucher-ticket)

## 模板消息

### [addTemplate](https://uniapp.dcloud.io/api/other/template?id=addtemplate)

组合模板并添加至帐号下的个人模板库。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | x |

**请求地址**

/\* 微信小程序 \*/

POST https://api.weixin.qq.com/cgi-bin/wxopen/template/add?access\_token=ACCESS\_TOKEN

/\* 百度小程序 \*/

POST https://openapi.baidu.com/rest/2.0/smartapp/template/templateadd?access\_token=ACCESS\_TOKEN

**请求参数**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| access\_token | string | 是 | 接口调用凭证，微信小程序参考 [getAccessToken](https://developers.weixin.qq.com/miniprogram/dev/api/getAccessToken.html" \t "_blank)，百度小程序参考[verify](https://smartprogram.baidu.com/docs/develop/api/open_userinfo/#verify/) |
| id | string | 是 | 模板标题id |
| keyword\_id\_list | Array.<number> | 是 | 开发者自行组合好的模板关键词列表，关键词顺序可以自由搭配（例如[3,5,4]或[4,5,3]），最多支持10个关键词组合 |

**返回值(Object)**

| **属性** | **类型** | **说明** | **平台** |
| --- | --- | --- | --- |
| errcode | number | 错误码 |  |
| errmsg | string | 错误信息 |  |
| template\_id | string | 添加至帐号下的模板id，发送APP模板消息时所需 | 微信小程序 |
| data | Object | {template\_id} | 百度小程序 |

**Tips**

* POST 数据格式：JSON。
* access\_token 应写在 url 上。
* 本接口应在后端服务器调用。

**请求数据示例**

{

"id": "xxx",

"keyword\_id\_list": [3, 4, 5]

}

**返回数据示例**

{

"errcode": 0,

"errmsg": "ok",

/\* "template\_id": "wDYzYZVxobJivW9oMpSCpuvACOfJXQIoKUm0PY397Tc" //微信小程序 \*/

"data": { // 百度小程序

"template\_id": "f34178cd598201d9dc8d5c88cd87b44cf7cd0e62NwmP"

}

}

### [deleteTemplate](https://uniapp.dcloud.io/api/other/template?id=deletetemplate)

删除帐号下的某个模板。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | x |

**请求地址**

/\* 微信小程序 \*/

POST https://api.weixin.qq.com/cgi-bin/wxopen/template/del?access\_token=ACCESS\_TOKEN

/\* 百度小程序 \*/

POST https://openapi.baidu.com/rest/2.0/smartapp/template/templatedel?access\_token=ACCESS\_TOKEN

**请求参数**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| access\_token | string | 是 | 接口调用凭证，微信小程序参考 [getAccessToken](https://developers.weixin.qq.com/miniprogram/dev/api/getAccessToken.html" \t "_blank)，百度小程序参考[verify](https://smartprogram.baidu.com/docs/develop/api/open_userinfo/#verify/) |
| template\_id | string | 是 | 要删除的模板id |

**返回值（Object）**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| errcode | number | 错误码 |
| errmsg | string | 错误信息 |

**Tips**

* POST 数据格式：JSON。
* access\_token 应写在 url 上。
* 本接口应在后端服务器调用。

**请求数据示例**

{

"template\_id": "wDYzYZVxobJivW9oMpSCpuvACOfJXQIoKUm0PY397Tc"

}

**返回数据示例**

{

"errcode": 0,

"errmsg": "ok"

}

### [getTemplateLibraryById](https://uniapp.dcloud.io/api/other/template?id=gettemplatelibrarybyid)

获取模板库某个模板标题下关键词库。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | x |

**请求地址**

/\* 微信小程序 \*/

POST https://api.weixin.qq.com/cgi-bin/wxopen/template/library/get?access\_token=ACCESS\_TOKEN

/\* 百度小程序 \*/

POST https://openapi.baidu.com/rest/2.0/smartapp/template/libraryget?access\_token=ACCESS\_TOKEN

**请求参数**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| access\_token | string | 是 | 接口调用凭证，微信小程序参考 [getAccessToken](https://developers.weixin.qq.com/miniprogram/dev/api/getAccessToken.html" \t "_blank)，百度小程序参考[verify](https://smartprogram.baidu.com/docs/develop/api/open_userinfo/#verify/) |
| id | string | 是 | 模板标题id |

**返回值(Object)**

| **属性** | **类型** | **说明** | **平台** |
| --- | --- | --- | --- |
| errcode | number | 错误码 | 微信小程序 |
| errmsg | string | 错误信息 | 微信小程序 |
| id | string | 模板标题 id | 微信小程序 |
| title | string | 模板标题 | 微信小程序 |
| keyword\_list | Array.<Object> | 关键词列表 | 微信小程序 |
| errno | number | 错误码 | 百度小程序 |
| msg | string | 错误信息 | 百度小程序 |
| data | Object | {id,title,keyword\_count,keyword\_list} | 百度小程序 |

**keyword\_list 的结构**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| keyword\_id | string | 关键词 id，添加模板时需要 |
| name | string | 关键词内容 |
| example | string | 关键词内容对应的示例 |

**Tips**

* POST 数据格式：JSON。
* access\_token 应写在 url 上。
* 本接口应在后端服务器调用。

**请求数据示例**

{

"id": "xxx"

}

**微信小程序返回数据示例**

{

"errcode": 0,

"errmsg": "ok",

"id": "AT0002",

"title": "购买成功通知",

"keyword\_list": [

{

"keyword\_id": 3,

"name": "购买地点",

"example": "TIT造舰厂"

},

{

"keyword\_id": 4,

"name": "购买时间",

"example": "2016年6月6日"

}

]

}

**百度小程序返回数据示例**

{

"errno": 0,

"msg": "success",

"data": {

"id": "BD0016",

"title": "取票成功通知",

"keyword\_count": 13,

"keyword\_list": [

{

"keyword\_id": 1,

"name": "金额",

"example": "300元"

},

{

"keyword\_id": 2,

"name": "订单号",

"example": "321254555"

}

]

}

}

### [getTemplateLibraryList](https://uniapp.dcloud.io/api/other/template?id=gettemplatelibrarylist)

获取APP模板库标题列表

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | x |

**请求地址**

/\* 微信小程序 \*/

POST https://api.weixin.qq.com/cgi-bin/wxopen/template/library/list?access\_token=ACCESS\_TOKEN

/\* 百度小程序 \*/

POST https://openapi.baidu.com/rest/2.0/smartapp/template/librarylist?access\_token=ACCESS\_TOKEN

**请求参数**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| access\_token | string | 是 | 接口调用凭证，微信小程序参考 [getAccessToken](https://developers.weixin.qq.com/miniprogram/dev/api/getAccessToken.html" \t "_blank)，百度小程序参考[verify](https://smartprogram.baidu.com/docs/develop/api/open_userinfo/#verify/) |
| offset | number | 是 | 用于分页，表示从offset开始。从 0 开始计数。 |
| count | number | 是 | 用于分页，表示拉取count条记录。最大为 20。 |

**返回值(Object)**

| **属性** | **类型** | **说明** | **平台** |
| --- | --- | --- | --- |
| errcode | number | 错误码 | 微信小程序 |
| errmsg | string | 错误信息 | 微信小程序 |
| list | Array.<Object> | 返回模板库标题列表 | 微信小程序 |
| total\_count | number | 模板库标题总数 | 微信小程序 |
| errno | number | 错误码 | 百度小程序 |
| msg | string | 错误信息 | 百度小程序 |
| data | Object | {total\_count,list} | 百度小程序 |

**list结构说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| id | string | 模板标题id（获取模板标题下的关键词库时需要） |
| title | string | 模板标题内容 |

**Tips**

* POST 数据格式：JSON。
* access\_token 应写在 url 上。
* 本接口应在后端服务器调用。

**请求数据示例**

{

"offset": 0,

"count": 5

}

**微信小程序返回数据示例**

{

"errcode": 0,

"errmsg": "ok",

"list": [

{"id": "AT0002", "title": "购买成功通知"},

{"id": "AT0003", "title": "交易提醒"}

],

"total\_count": 100

}

**百度小程序返回数据示例**

{

"errno": 0,

"msg": "success",

"data": {

"total\_count": 100,

"list": [

{"id": "AT0002", "title": "购买成功通知"},

{"id": "AT0003", "title": "交易提醒"}

]

}

}

### [getTemplateList](https://uniapp.dcloud.io/api/other/template?id=gettemplatelist)

获取帐号下已存在的模板列表。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | x |

**请求地址**

/\* 微信小程序 \*/

POST https://api.weixin.qq.com/cgi-bin/wxopen/template/list?access\_token=ACCESS\_TOKEN

/\* 百度小程序 \*/

POST https://openapi.baidu.com/rest/2.0/smartapp/template/templatelist?access\_token=ACCESS\_TOKEN

**请求参数**

| **属性** | **类型** | **必填** | **说明** |
| --- | --- | --- | --- |
| access\_token | string | 是 | 接口调用凭证，微信小程序参考 [getAccessToken](https://developers.weixin.qq.com/miniprogram/dev/api/getAccessToken.html" \t "_blank)，百度小程序参考[verify](https://smartprogram.baidu.com/docs/develop/api/open_userinfo/#verify/) |
| offset | number | 是 | 用于分页，表示从offset开始。从 0 开始计数。 |
| count | number | 是 | 用于分页，表示拉取count条记录。最大为 20。 |

**返回值(Object)**

| **属性** | **类型** | **说明** | **平台** |
| --- | --- | --- | --- |
| errcode | number | 错误码 | 微信小程序 |
| errmsg | string | 错误信息 | 微信小程序 |
| list | Array.<Object> | 返回模板列表 | 微信小程序 |
| errno | number | 错误码 | 百度小程序 |
| msg | string | 错误信息 | 百度小程序 |
| data | Object | {total\_count, list} | 百度小程序 |

**list结构说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| template\_id | string | 添加至帐号下的模板id，发送APP模板消息时所需 |
| title | string | 模板标题 |
| content | string | 模板内容 |
| example | string | 模板内容示例 |

**Tips**

* POST 数据格式：JSON。
* access\_token 应写在 url 上。
* 本接口应在后端服务器调用。

**请求数据示例**

{

"offset": 0,

"count": 1

}

**微信小程序返回数据示例**

{

"errcode": 0,

"errmsg": "ok",

"list": [

{

"template\_id": "wDYzYZVxobJivW9oMpSCpuvACOfJXQIoKUm0PY397Tc",

"title": "购买成功通知",

"content": "购买地点{{keyword1.DATA}}\n购买时间{{keyword2.DATA}}\n物品名称{{keyword3.DATA}}\n",

"example": "购买地点：TIT造舰厂\n购买时间：2016年6月6日\n物品名称：咖啡\n"

}

]

}

**百度小程序返回数据示例**

{

"errno": 0,

"msg": "success",

"data": {

"total\_count": 1,

"list": [

{

"template\_id": "e4313219538c4b0262e3a14a0507000e8bd79e9PTPAz",

"title": "续费成功通知",

"content": "购买时间{{keyword1.DATA}}\n物品名称{{keyword2.DATA}}",

"example": "购买时间: 2016年6月6日\n物品名称: 奶茶"

}

]

}

}

### [sendTemplateMessage](https://uniapp.dcloud.io/api/other/template?id=sendtemplatemessage)

发送模板消息

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | x |

**请求地址**

/\* 微信小程序 \*/

POST https://api.weixin.qq.com/cgi-bin/message/wxopen/template/send?access\_token=ACCESS\_TOKEN

/\* 百度小程序 \*/

POST https://openapi.baidu.com/rest/2.0/smartapp/template/templatedel?access\_token=ACCESS\_TOKEN

**请求参数**

| **属性** | **类型** | **必填** | **说明** | **支持平台** |
| --- | --- | --- | --- | --- |
| access\_token | string | 是 | 接口调用凭证，微信小程序参考 [getAccessToken](https://developers.weixin.qq.com/miniprogram/dev/api/getAccessToken.html" \t "_blank)，百度小程序参考[verify](https://smartprogram.baidu.com/docs/develop/api/open_userinfo/#verify/) |  |
| touser | string | 是 | 接收者（用户）的 openid/swan\_id |  |
| template\_id | string | 是 | 所需下发的模板消息的id |  |
| page | string | 否 | 点击模板卡片后的跳转页面，仅限本APP内的页面。支持带参数,（示例index?foo=bar）。该字段不填则模板无跳转。 |  |
| data | Object | 否 | 模板内容，不填则下发空模板。具体格式请参考示例。 |  |
| form\_id | string | 是 | 表单提交场景下，为 submit 事件带上的 [formId](https://uniapp.dcloud.io/component/form)；支付场景下，为本次支付的 [prepay\_id](https://uniapp.dcloud.io/api/plugins/payment?id=requestpayment) | 微信小程序 |
| emphasis\_keyword | string | 否 | 模板需要放大的关键词，不填则默认无放大 | 微信小程序 |
| touser\_openId | string | 是 | 接收者open\_id | 百度小程序 |
| scene\_id | string | 是 | string | 百度小程序 |
| scene\_type | number | 是 | 场景type，1：表单；2：百度收银台订单；3:直连订单。 | 百度小程序 |
| ext | jsonString | 否 | '{"xzh\_id":111,"category\_id":15}' | 百度小程序 |

**返回值(Object)**

| **属性** | **类型** | **说明** | **平台** |
| --- | --- | --- | --- |
| errcode | number | 错误码 | 微信小程序 |
| errmsg | string | 错误信息 | 微信小程序 |
| errno | number | 错误码 | 百度小程序 |
| msg | string | 错误信息 | 百度小程序 |
| data | Object | {msg\_key} | 百度小程序 |

**errcode 的合法值**

| **值** | **说明** |
| --- | --- |
| 40037 | template\_id不正确 |
| 41028 | form\_id不正确，或者过期 |
| 41029 | form\_id已被使用 |
| 41030 | page不正确 |
| 45009 | 接口调用超过限额（目前默认每个帐号日调用限额为100万） |

**errno 的合法值**

| **值** | **说明** |
| --- | --- |
| 2002 | 参数错误 |
| 4001 | template\_id 不正确 |
| 4002 | 消息推送接口调用失败 |
| 4003 | 表单无效 |
| 4004 | 场景id无效 |
| 6001 | 无 push 权限 |

**Tips**

* POST 数据格式：JSON。
* access\_token 应写在 url 上。
* 本接口应在后端服务器调用。

**微信小程序请求数据示例**

{

"touser": "OPENID",

"template\_id": "TEMPLATE\_ID",

"page": "/pages/index/index",

"form\_id": "FORMID",

"data": {

"keyword1": {

"value": "339208499"

},

"keyword2": {

"value": "2015年01月05日 12:30"

}

},

"scene\_type": 1,

"emphasis\_keyword": "keyword1.DATA"

}

**微信小程序返回数据示例**

{

"errcode": 0,

"errmsg": "ok"

}

**百度小程序请求数据示例**

{

"touser": "SWAN\_ID",

"template\_id": "TEMPLATE\_ID",

"page": "/pages/index/index",

"data": {

"keyword1": {

"value": "339208499"

},

"keyword2": {

"value": "2015年01月05日 12:30"

}

},

"scene\_id": "123456"

}

**百度小程序返回数据示例**

{

"errno": 0,

"msg": "success",

"data": {

"msg\_key": 158

}

}

### [alipay.open.app.mini.templatemessage.send](https://uniapp.dcloud.io/api/other/template?id=alipayopenappminitemplatemessagesend)

小程序通过 openapi 给用户触达消息，主要为支付后的触达（通过消费id）和用户提交表单后的触达（通过formId）。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | x | √ | x | x | x |

**请求地址**

https://openapi.alipay.com/gateway.do

**公共请求参数**

| **名称** | **类型** | **必填** | **描述** | **示例值** |
| --- | --- | --- | --- | --- |
| app\_id | String | 是 | 支付宝分配给开发者的应用ID | 2014072300007148 |
| method | String | 是 | 接口名称 | alipay.open.app.mini.templatemessage.send |
| format | String | 是 | 仅支持JSON | JSON |
| charset | String | 是 | 请求使用的编码格式，如utf-8,gbk,gb2312等 | utf-8 |
| sign\_type | String | 是 | 商户生成签名字符串所使用的签名算法类型，目前支持RSA2 | RSA2 |
| sign | String | 是 | 商户请求参数的签名串，详见签名 | 详见示例 |
| timestamp | String | 是 | 发送请求的时间，格式 yyyy-MM-dd HH:mm:ss | 2014-07-24 03:07:50 |
| version | String | 是 | 调用的接口版本，固定为：1.0 | 1.0 |
| app\_auth\_token | String | 否 | 详见[应用授权概述](https://doc.open.alipay.com/doc2/detail.htm?treeId=216&articleId=105193&docType=1) |  |
| biz\_content | String | 是 | 请求参数的集合，最大长度不限，除公共参数外所有请求参数都必须放在这个参数中传递，具体参照各产品快速接入文档 |  |

## 订阅消息

### [uni.requestSubscribeMessage(Object object)](https://uniapp.dcloud.io/api/other/requestSubscribeMessage?id=requestsubscribemessage)

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | 基础库版本>=2.8.2 | x | x | x | x |

**object参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| tmplIds | Array |  | 是 | 需要订阅的消息模板的id的集合，一次调用最多可订阅3条消息（注意：iOS客户端7.0.6版本、Android客户端7.0.7版本之后的一次性订阅/长期订阅才支持多个模板消息，iOS客户端7.0.5版本、Android客户端7.0.6版本之前的一次订阅只支持一个模板消息）消息模板id在[微信公众平台(mp.weixin.qq.com)-功能-订阅消息]中配置 |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**注意**

* 用户发生点击行为或者发起支付回调后，才可以调起订阅消息界面

**平台说明**

* [微信小程序订阅消息参考文档](https://developers.weixin.qq.com/miniprogram/dev/framework/open-ability/subscribe-message.html)

**object.success 返回值**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 接口调用成功时errMsg值为'requestSubscribeMessage:ok' |
| TEMPLATE\_ID | String | [TEMPLATE\_ID]是动态的键，即模板id，值包括'accept'、'reject'、'ban'。'accept'表示用户同意订阅该条id对应的模板消息，'reject'表示用户拒绝订阅该条id对应的模板消息，'ban'表示已被后台封禁。例如 { errMsg: "requestSubscribeMessage:ok", zun-LzcQyW-edafCVvzPkK4de2Rllr1fFpw2A\_x0oXE: "accept"} 表示用户同意订阅zun-LzcQyW-edafCVvzPkK4de2Rllr1fFpw2A\_x0oXE这条消息 |

**object.fail 返回值**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| errMsg | String | 接口调用失败错误信息 |
| errCode | Number | 接口调用失败错误码 |

**示例代码**

uni.requestSubscribeMessage({

tmplIds: [''],

success (res) { }

})

## 小程序更新

### [uni.getUpdateManager()](https://uniapp.dcloud.io/api/other/update?id=getupdatemanager)

本API返回**全局唯一**的版本更新管理器对象： updateManager，用于管理小程序更新。

App的更新不使用本API，另见文档：

* 整包更新：<https://ask.dcloud.net.cn/article/34972>
* 资源文件热更新(wgt升级)：[https://ask.dcloud.net.cn/article/35667](https://ask.dcloud.net.cn/article/35667" \t "_blank)

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | √ | √ | √ | √ |

**updateManager 对象的方法列表：**

| **方法** | **参数** | **说明** |
| --- | --- | --- |
| onCheckForUpdate | callback | 当向小程序后台请求完新版本信息，会进行回调 |
| onUpdateReady | callback | 当新版本下载完成，会进行回调 |
| onUpdateFailed | callback | 当新版本下载失败，会进行回调 |
| applyUpdate |  | 当新版本下载完成，调用该方法会强制当前小程序应用上新版本并重启 |

**onCheckForUpdate(callback) 回调结果说明：**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| hasUpdate | Boolean | 是否有新的版本 |

**代码示例**

const updateManager = uni.getUpdateManager();

updateManager.onCheckForUpdate(function (res) {

// 请求完新版本信息的回调

console.log(res.hasUpdate);

});

updateManager.onUpdateReady(function (res) {

uni.showModal({

title: '更新提示',

content: '新版本已经准备好，是否重启应用？',

success(res) {

if (res.confirm) {

// 新的版本已经下载好，调用 applyUpdate 应用新版本并重启

updateManager.applyUpdate();

}

}

});

});

updateManager.onUpdateFailed(function (res) {

// 新的版本下载失败

});

## 调试

### [uni.setEnableDebug(OBJECT)](https://uniapp.dcloud.io/api/other/set-enable-debug?id=setenabledebug)

设置是否打开调试开关。此开关对正式版也能生效。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | x | √ |

**OBJECT说明**

| **属性** | **类型** | **必填** | **说明** | **平台差异说明** |
| --- | --- | --- | --- | --- |
| enableDebug | boolean | 是 | 是否打开调试 |  |
| success | function | 否 | 接口调用成功的回调函数 | 微信小程序 |
| fail | function | 否 | 接口调用失败的回调函数 | 微信小程序 |
| complete | function | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） | 微信小程序 |

#### [示例代码](https://uniapp.dcloud.io/api/other/set-enable-debug?id=%e7%a4%ba%e4%be%8b%e4%bb%a3%e7%a0%81)

// 打开调试

uni.setEnableDebug({

enableDebug: true

})

// 关闭调试

uni.setEnableDebug({

enableDebug: false

})

## 获取第三方平台数据

### [uni.getExtConfig(OBJECT)](https://uniapp.dcloud.io/api/other/get-extconfig?id=getextconfig)

获取第三方平台自定义的数据字段。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | √ | x |

**OBJECT参数说明**

| **属性** | **类型** | **默认值** | **必填** | **说明** |
| --- | --- | --- | --- | --- |
| success | function |  | 否 | 接口调用成功的回调函数 |
| fail | function |  | 否 | 接口调用失败的回调函数 |
| complete | function |  | 否 | 接口调用结束的回调函数（调用成功、失败都会执行） |

**success 回调参数说明**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| extConfig | Object | 第三方平台自定义的数据 |

**Tips**

* 本接口暂时无法通过 uni.canIUse判断是否兼容，开发者需要自行判断 uni.getExtConfig 是否存在来兼容。

**示例代码**

if (uni.getExtConfig) {

uni.getExtConfig({

success(res) {

console.log(res.extConfig)

}

})

}

### [uni.getExtConfigSync()](https://uniapp.dcloud.io/api/other/get-extconfig?id=getextconfigsync)

uni.getExtConfig 的同步版本。

**平台差异说明**

| **App** | **H5** | **微信小程序** | **支付宝小程序** | **百度小程序** | **头条小程序** | **QQ小程序** |
| --- | --- | --- | --- | --- | --- | --- |
| x | x | √ | x | √ | √ | x |

**返回值(Object)**

| **属性** | **类型** | **说明** |
| --- | --- | --- |
| extConfig | Object | 第三方平台自定义的数据 |

**Tips**

* 本接口暂时无法通过 uni.canIUse判断是否兼容，开发者需要自行判断 uni.getExtConfigSync 是否存在来兼容。

**代码示例**

const extConfig = uni.getExtConfigSync ? uni.getExtConfigSync() : {}

console.log(extConfig)

## 广告

#### [广告](https://uniapp.dcloud.io/api/other/advertisement?id=%e5%b9%bf%e5%91%8a)

广告API

广告能力在不同小程序端实现不同，使用时需注意用[条件编译](https://uniapp.dcloud.io/platform)调用不同平台的代码。

* App平台：无需编码，在打包App时可直接勾选广告位，[详见](https://dcloud.io/dad.html)
* 微信小程序：[规范文档](https://developers.weixin.qq.com/miniprogram/dev/api/wx.createRewardedVideoAd.html)
* 百度小程序：有组件但无API
* 支付宝小程序：不支持此能力
* 头条小程序：仅小游戏可用，小程序不可用，不适用于uni-app
* QQ小程序：[规范文档](https://q.qq.com/wiki/develop/miniprogram/API/ad/qq.createRewardedVideoAd.html)