

酷开系统  
浏览器 **Native** 接口文档  
V0.1

# 修改记录

版本号	修改说明	修改日期	修改人
V0.1	初版发布	2019.5.24	

# 目录

目录.....	3
一、概述.....	4
1.1 Web 框架介绍.....	4
酷开 Web 开发框图参考.....	4
示例一：获取机器设备信息.....	5
示例二：跳转指定版面.....	7
1.2 JS-SDK 层内部封装说明.....	9
exec()方法.....	9
startapp 说明.....	9
事件注册接口.....	13
二、接口定义.....	16
2.1 系统接口.....	16
2.2 网络接口.....	19
2.3 下载接口.....	21
2.4 多媒体接口.....	24
2.5 设备信息.....	25
2.6 用户信息.....	29
2.7 日志接口.....	33
2.8 支付接口.....	34
2.9 优惠券接口.....	37
2.10 广告接口.....	39
2.11 影视模块.....	44
2.12 商城模块.....	47
2.13 应用模块.....	50
2.14 游戏模块.....	52
2.15 启动 Web 页接口.....	54
2.16 页面生命周期及事件.....	54
2.17 安卓小程序接口.....	56
2.18 广播接口.....	56

# 一、概述

酷开平台 Web 开发接口，通过实现这套接口，支持酷开现有的 Web 页面，实现酷开系统无缝切换到第三方浏览器的目标。

本文档章节简要说明：

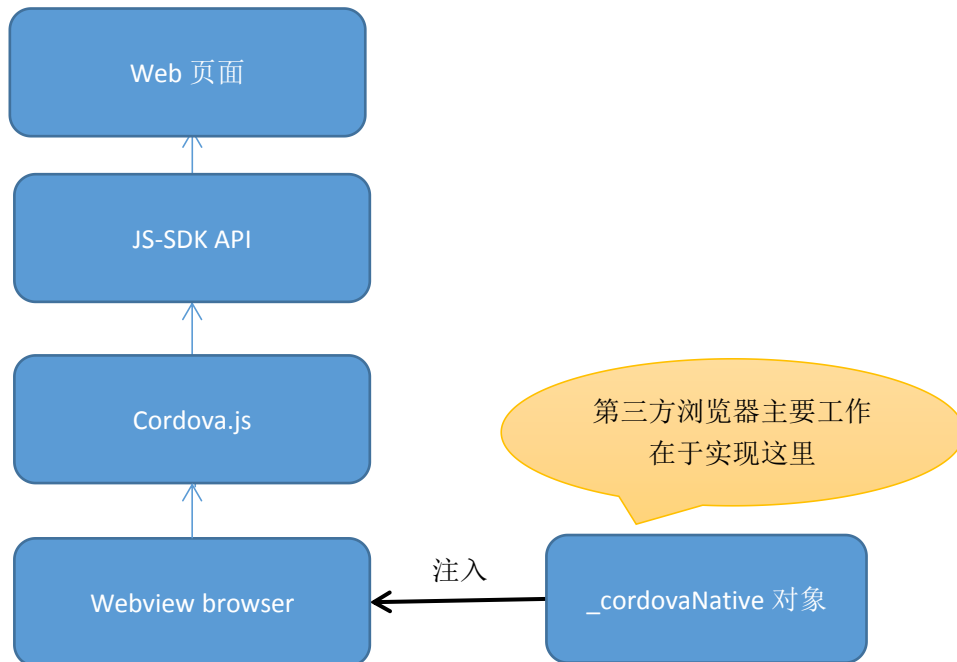
第一章“Web 框架介绍”章节简单说明了酷开系统使用的 Web 开发系统，并用两个实例辅助理解；

第二章“JS-SDK 层内部封装说明”对第三章的接口详细说明做了背景铺垫；

第三章“接口定义”分模块详细说明了目前使用的接口。

## 1.1 Web 框架介绍

### 酷开 Web 开发框图参考



第三方浏览器主要工作在：实现注入到浏览器的 native 层接口。

以下两个示例供参考：

## 示例一：获取机器设备信息

### Step 1. 在 Web 页面，调用接口获取设备信息：

（接口详情请参考‘接口定义-设备信息-coocaaosapi.getDeviceInfo’章节的说明）

```
coocaaosapi.getDeviceInfo(  
  success(res) {  
    console.log(res.panel);  
    console.log(res.chip);  
    console.log(res.activeid);  
    ...  
  }  
)
```

正确时返回值如下：

```
{  
  "panel": "32",  
  "chip": "8S47",  
  "chipid": "MST-6A338",  
  "sid": "ebc9f795-788e-4c8f-b0cc-cd5371e112f3",  
  "model": "E2A",  
  "devid": "c86ae6c2352dd8b8d6624c6af6d8a5ad",  
  "emmcid": "11010030303447363002129d2fa2a400",  
  "activeid": "31140974",  
  "brand": "Skyworth",  
  "barcode": "32E2AXX-S093662-Z180516-406D9",  
  "mac": "2835452aa239",  
  "androidsdk": 19,  
  "version": "7.00.190222"  
}
```

### Step 2. JS-SDK API 层：

//获取设备信息

```
CoocaaOSApi.prototype.getDeviceInfo= function(success, error) {  
  argscheck.checkArgs('ff', 'CoocaaOSApi.getBaseInfo', arguments);  
  exec(success, error, 'CoocaaOSApi', 'getDeviceInfo', []);  
}
```

### Step 3. Cordova.js 层：

在 cordova.js 里完成 JS 层 exec()方法到 Native 层\_cordovaNative 对象 exec()方法的参数转换：

```
function androidExec(success, fail, service, action, args) {  
  //省略部分代码...  
  // Process any ArrayBuffers in the args into a string.  
  for (var i = 0; i < args.length; i++) {
```

```
        if (utils.typeName(args[i]) == 'ArrayBuffer') {
            args[i] = base64.fromArrayBuffer(args[i]);
        }
    }

    var callbackId = service + cordova.callbackId++,
        argsJson = JSON.stringify(args);

    if (success || fail) {
        cordova.callbacks[callbackId] = {
            success: success,
            fail: fail
        };
    }

    //完成参数转换:
    var msgs = nativeApiProvider.get().exec(bridgeSecret, service, action,
                                                callbackId, argsJson);

    ...//省略部分代码
}
```

### Step 4. Native 层 \_cordovaNative 对象:

调用到 Native 层 `_cordovaNative` 对象 `exec()` 方法。

```

/*
 * Any exposed Javascript API MUST implement these three things!
 */

public interface ExposedJsApi {
    public String exec(int bridgeSecret, String service, String action, String callbackId,
String arguments) throws JSONException, IllegalAccessException;
    public String retrieveJsMessages(int bridgeSecret, boolean fromOnlineEvent) throws
IllegalAccessException;
}

Public class _cordovaNative implements ExposedJsApi{
    public String exec(int bridgeSecret, String service, String action, String callbackId,
String arguments) throws JSONException, IllegalAccessException
    {
        If(service == 'CoocaaOSApi'){
            Switch(action) {
                Case :'getDeviceInfo':
                    //获取各个信息，并封装到 info 对象
                    If(ok) {
                        callback.success(info)
                    }else {
                        callback.error("json params: " + e.toString());
                    }
            }
        }
    }
}

```

```

        Break;
    }
}
}
}
}

```

## 示例二：跳转指定版面

### Step 1. 在 Web 页面，调用打开‘电视信号源’接口：

```

coocaaosapi.startSourceList(
    function(message) { console.log(message); },
    function(error) { console.log(error); }
);

```

### Step 2. JS-SDK API 层：

```

/*
 *启动信号源
 */
CoocaaOSApi.prototype.startSourceList = function(success, error) {
    argscheck.checkArgs('ff', 'CoocaaOSApi.startSourceList', arguments);
    exec(success, error, 'CoocaaOSApi', 'launchSourceList', []);
}

```

### Step 3. Cordova.js 层：

在 cordova.js 里完成 JS 层 exec() 方法到 Native 层 \_cordovaNative 对象 exec() 方法的参数转换：

```

function androidExec(success, fail, service, action, args) {
    //省略部分代码...
    // Process any ArrayBuffers in the args into a string.
    for (var i = 0; i < args.length; i++) {
        if (utils.typeName(args[i]) == 'ArrayBuffer') {
            args[i] = base64.fromArrayBuffer(args[i]);
        }
    }
    var callbackId = service + cordova.callbackId++,
        argsJson = JSON.stringify(args);

    if (success || fail) {
        cordova.callbacks[callbackId] = {

```

```

        success: success,
        fail: fail
    };
}
//完成参数转换:
var msgs = nativeApiProvider.get().exec(bridgeSecret, service, action,
                                         callbackId, argsJson);

...//省略部分代码
}

```

#### Step 4. Native 层 \_cordovaNative 对象:

调用到 Native 层 \_cordovaNative 对象 exec() 方法。

```

/*
 * Any exposed Javascript API MUST implement these three things!
 */
public interface ExposedJsApi {
    public String exec(int bridgeSecret, String service, String action, String callbackId,
String arguments) throws JSONException, IllegalAccessException;
    public String retrieveJsMessages(int bridgeSecret, boolean fromOnlineEvent) throws
IllegalAccessException;
}

Public class _cordovaNative implements ExposedJsApi{
    public String exec(int bridgeSecret, String service, String action, String callbackId,
String arguments) throws JSONException, IllegalAccessException
    {
        If(service == 'CoocaaOSApi'){
            Switch(action) {
                Case : 'launchSourceList':
                    //跳转到信号源版面
                    If(ok) {
                        callback.success()
                    }else {
                        callback.error("json params: " + e.toString());
                    }
                    Break;
            }
        }
    }
}
}

```

第三方浏览器的主要工作在 cordovaNative 对象 exec() 和 retrieveJsMessages() 接口的实现。



## 1.2 JS-SDK 层内部封装说明

为了方便调用，JS-SDK 层在 `exec()` 基础上又简单封装了几个方法，以更方便简洁地实现各个接口；

部分接口实现时会用此处封装的方法，部分接口则直接调用 `exec()` 方法。

本质上 JS-SDK 层所有接口最终都是通过调用 `cordova` 框架提供的 `exec()` 方法与 Native 层通信的，

下面分别说明封装的方法：

### `exec()` 方法

这里针对 JS-SDK API 层用到的 `exec()` 方法进行说明，在 Native 层实现时请参考以上 step3 进行转换即可。

方法原型：

```
androidExec(success, fail, service, action, args)
```

参数类型：

参数	类型	默认值	必填	说明
success	Function	null	否	接口调用成功的回调函数
fail	function	null	否	接口调用失败的回调函数
service	String	-	是	
action	String	-	是	
args	Array.<String> 字符串数组	-	否	参数格式详见具体接口；

success 回调函数 参数

属性	类型	说明
Message	String/Object	Success 回调函数参数默认值是字符串“OK”，部分接口根据需要返回 Object 对象；

fail 回调函数 参数

属性	类型	说明
Message	String	返回错误说明字符串

Native 层根据接收到的 `service` 和 `action` 参数值，确定具体执行的操作。

### startapp 说明

在 JS-SDK 层对 `exec()` 又封装了一层，封装成 `startapp` 对象，方便部分接口代码复用，实现 JS-SDK 层时，部分接口用 `startapp.start()` 方法，部分接口直接用 `exec()` 方法；

如下代码可以看出，`startapp` 对象与 `exec()` 方法仅是简单的调用格式封装，下面章节中接口详细说明时会根据该接口调用的是 `startapp` 对象方法还是 `exec()` 方法，直接说明，不做转换。

```
startapp = {
  check: function(message, completeCallback, errorCallback) {
    exec(completeCallback, errorCallback, "startApp", "check", [message]);
  },
  start: function(message, completeCallback, errorCallback) {
    exec(completeCallback, errorCallback, "startApp", "start",
      (typeof message === 'string') ? [message] : message);
  },
  play: function(message, completeCallback, errorCallback) {
    exec(completeCallback, errorCallback, "startApp", "play",
      (typeof message === 'string') ? [message] : message);
  },
  startservice: function(message, completeCallback, errorCallback) {
    exec(completeCallback, errorCallback, "startApp", "startservice",
      (typeof message === 'string') ? [message] : message);
  }
}
```

Startapp 对象的各方法简介：

## startapp.check

检测指定包名的应用信息

```
check: function(message, completeCallback, errorCallback) {
  exec(completeCallback, errorCallback, "startApp", "check", [message]);
},
```

message 的合法值

值	说明
com.coocaa.app_browser	酷开新版浏览器
com.tianci.user	账户应用
com.coocaa.activecenter	活动中心
com.coocaa.mall	酷开商城
com.tianci.movieplatform	酷开影视
com.tianci.localmedia	本地媒体
com.tianci.setting	电视设置
...	其它 apk 包名...

success 回调函数 参数

	类型	说明
Message	Object	参考如下示例

fail 回调函数 参数

	类型	说明
Message	String	返回错误说明字符串

示例：

```
startapp.check("com.coocaa.app_browser", function(message) { /* success */
    console.log("新版浏览器存在： " + JSON.stringify(message));
    var browserVersion = message.versionCode
}, function(message) {
    console.log("新版浏览器不存在： " + JSON.stringify(message));
});
```

Startapp.check 接口正确时的返回值：

```
{
    "packageName": "com.coocaa.app_browser",
    "versionCode": 200036,
    "applicationInfo": "ApplicationInfo{4216b518 com.coocaa.app_browser}",
    "versionName": "2.0.36"
}
```

出错时返回错误信息：

```
android.content.pm.PackageManager$NameNotFoundException:
com.coocaa.app_browsertesttest"
```

## startapp.start

打开指定包名的应用

```
start: function(message, completeCallback, errorCallback) {
    exec(completeCallback, errorCallback, "startApp", "start",
        (typeof message === 'string') ? [message] : message);
}
```

message 参数格式说明：

startapp.start([[param1,param2,param3,param4,param5],str], success, error);

根据不同情况会有如下格式，都需支持：

message 数组第 1 个元素：可以是一个字符串，也可以是一个数组：

Case 字符串：参考以下表格 case 1；

Case 数组：

数组最多有 5 个参数，取值情况为：

- 1) 如果用 activity 方式启动：1,2 传参为包名、类名；3、4、5 为空；
- 2) 如果用其他方式启动，1 传参"action",2\3\4\5 可选---2action 名、Suri 地址

message 数组第 2 个元素: str 为可选的拓展参数，格式是对象数组：

[{key1:"value1"},{key2:"value2"}]

以上情况举例如下：

	格式	说明	message 示例
第一个元素是字符串时	String	以包名方式启动应用	'com.tianci.localmedia'
第一个元素是数组时	Array<Array<String>> 二维数组	以包名加类名方式启动应用	[[ "com.tianci.localmedia", "com.tianci.localmedia.MainActivity" ]]
有第二个元素时	Array<Array<String>, Array<Object>>	以包名加类名方式启动应用，并附带应用参数。	[[ "action", "android.settings.ADD_ACCOUNT_SETTINGS"], [{ 'needFinish': true }, { 'type': type }]]

**object.success** 回调函数 参数

属性	类型	说明
Message	String	返回字符串 “OK”

**object.fail** 回调函数 参数

属性	类型	说明
Message	String	返回错误说明字符串

示例：

case 1:

```
startapp.start("com.tianci.localmedia", success, error)
```

Case 2:

```
startapp.start([ "com.tianci.localmedia", "com.tianci.localmedia.MainActivity" ],  
               success, error);
```

Case 3:

```
startapp.start([ "action", "android.settings.ADD_ACCOUNT_SETTINGS",  
                 [{ 'needFinish': true }, { 'type': type } ] ], success, error);
```

**startapp.startservice**

打开指定包名的服务

```
startservice: function(message, completeCallback, errorCallback) {  
    exec(completeCallback, errorCallback, "startApp", "startservice",  
        (typeof message === 'string') ? [message] : message);  
}
```

**message** 参数格式及回调同 **startapp.start** 方法。

**startapp.play**

打开指定包名的应用（暂时没用到？）

```
play: function(message, completeCallback, errorCallback) {
```

```
exec(completeCallback, errorCallback, "startApp", "play",
      (typeof message === 'string') ? [message] : message);
},
```

message 参数格式说明:

	格式	说明	示例
1	Array<String, String,Array<Object>>	Action+ 包名+ 需要时附带应用参数	[ "action", "coocaa.intent.movie.videospecial", [{'topicCode': detailid}] ]

object.success 回调函数 参数

属性	类型	说明
Message	String	返回字符串 "OK"

object.fail 回调函数 参数

属性	类型	说明
Message	String	返回错误说明字符串

## 事件注册接口

在 JS-SDK 层的事件注册函数，将对 exec() 的调用封装成 broadcaster 对象，调用更简洁方便；另外回调函数统一在 JS-SDK 层处理，Native 层只用处理事件注册即可。

## addEventListener

注册指定事件监听

```
addEventListener: function(eventname, f) {
    exec(function() {
        me._channels[eventname] = channel.create(eventname);
        me._channels[eventname].subscribe(f);
    }, function(err) {
        console.log("ERROR addEventListener: " + err)
    }, "broadcaster", "addEventListener", [eventname]);
    ...
}
```

eventname 参数格式说明:

	格式	说明	示例
1	String	事件名称	"NET_CHANGGED"

eventname 的合法值

值	说明
NET_CHANGGED	网络状态变化事件
USB_CHANGGED	USB 设备状态变化事件
APP_TASK_CALLBACK	APK 下载事件监听
USER_CHANGGED	用户登录状态变化
PURCHASE_CALLBACK	订单状态变化
COMMON_CHANGED	通用监听

success 回调函数 参数

属性	类型	说明
Message	String	返回字符串 “OK”

fail 回调函数 参数

属性	类型	说明
Message	String	返回错误说明字符串

## removeEventListener

删除事件监听函数：

```
removeEventListener: function(eventname, f) {  
    if (eventname in this._channels) {  
        exec(function() {}, function(err) {  
            console.log("ERROR removeEventListener: " + err)  
        }, "broadcaster", "removeEventListener", [eventname]);  
    }  
}
```

eventname 参数格式说明，取值参考 addEventListener

	格式	说明	示例
1	String	事件名称	“NET_CHANGGED”

object.success 回调函数 参数

属性	类型	说明
Message	String	返回字符串 “OK”

object.fail 回调函数 参数

属性	类型	说明
Message	String	返回错误说明字符串

## addGlobalEventListener

注册全局指定事件监听

```
addGlobalEventListener: function (eventname,f) {
```

```

        if (!(eventname in this._channels)) {
            var me = this;
            exec( function() {
                me._channels[eventname] = channel.create(eventname);
                me._channels[eventname].subscribe(f);
            }, function(err) {
                console.log( "ERROR addGlobalEventListener: " + err)
            }, "broadcaster", "addGlobalEventListener", [ eventname ]));
            //...
        }
    }
}

```

eventname 参数格式说明：

	格式	说明	示例
1	String	事件名称	“NET_CHANGGED”

object.success 回调函数 参数

属性	类型	说明
Message	String	返回字符串 “OK”

object.fail 回调函数 参数

属性	类型	说明
Message	String	返回错误说明字符串

## removeGlobalEventListener

删除全局事件监听

```

removeGlobalEventListener: function(eventname, f) {
    if (eventname in this._channels) {
        var me = this;
        exec( function() {
            me._channels[eventname].unsubscribe(f);
        }, function(err) {
            console.log( "ERROR removeGlobalEventListener: " + err)
        }, "broadcaster", "removeGlobalEventListener", [ eventname ]));
    }
}

```

eventname 参数格式说明：

	格式	说明	示例
1	String	事件名称	“NET_CHANGGED”

object.success 回调函数 参数

属性	类型	说明
Message	String	返回字符串 “OK”

object.fail 回调函数 参数

属性	类型	说明
Message	String	返回错误说明字符串

## 二、接口定义

下面是目前使用到的接口详细说明，分各个模块介绍，可能会有少部分接口尚未添加进来，后续版本持续完善。

### 2.1 系统接口

#### coocaaosapi.getAppInfo()

获取 app 版本信息

```
exec(success, error, 'CoocaaOSApi', 'getAppInfo', [{ 'pkgList': packageName }])
```

message 参数格式说明：

参数传递一个对象，key 为"pkgList",value 为应用包名的数组。

即{pkgList:["com.tianci.user","com.tianci.movieplatform"]}

object.success 回调函数

参数 Object res

属性	类型	说明
versionCode	Number	
Status	String	0:ok
versionName	String	

示例代码：

```
var packageName = '{"pkgList":["com.tianci.user","com.coocaa.mall"]}'
exec(success, error, 'CoocaaOSApi', 'getAppInfo', [{ 'pkgList': packageName }]);
```

返回值如下：

```
{\"com.coocaa.mall\":{\\\"versionCode\\\":31100003,\\\"status\\\":\\\"0\\\",\\\"versionName\\\":\\\"3.11.3[CC][1904030048]\\\",\\\"com.tianci.user\\\":{\\\"versionCode\\\":4100018,\\\"status\\\":\\\"0\\\",\\\"versionName\\\":\\\"4.10.18\\\"}}}
```

#### coocaaosapi.getPushInfo()

获取应用 push 信息

```
exec(success, error, 'CoocaaOSApi', 'getPushInfo', [{ 'pkgList': packageName }])
```



message 参数格式说明:

参数传递一个对象, key 为"pkgList",value 为应用包名的数组。

即{pkgList:["com.tianci.user","com.tianci.movieplatform"]}

object.success 回调函数

参数 Object res

属性	类型	说明
status	String	0:ok, -1: 没有
pushId	String	

示例代码:

```
var packageName = '{"pkgList":["com.tianci.user","com.coocaa.mall"]}'
exec(success, error, 'CoocaaOSApi', 'getAppInfo', [{ 'pkgList': packageName}]);
```

返回值如下:

```
{\"com.coocaa.mall\":{\\\"status\\\":\\\"-1\\\",\\\"pushId\\\":\\\"\\\",\\\"com.tianci.user\\\":{\\\"s
tatus\\\":\\\"0\\\",\\\"pushId\\\":\\\"8024f864706d485f935eb7e0b1cd9e99c903a65d\\\"}}
```

## coocaaosapi.getPropertiesValue()

获取属性值。

exec(success, error, 'CoocaaOSApi', 'getPropertiesValue', [{'propertiesKey': data}])

参数 data

属性	类型	默认值	必填	说明
Data	String	-	是	属性的 key

object.success 回调函数

参数 Object res

属性	类型	说明
propertiesValue	String	属性的值

示例代码:

```
exec(success, error, 'CoocaaOSApi', 'getPropertiesValue',
    [{ 'propertiesKey': "third.get.4k"}]);
```

返回值:

```
{"propertiesValue":"0"}
```

## coocaaosapi.addUSBChangedListener(function callback)

USB 状态事件监听

brocaster.addEventListener("USB\_CHANGGED", listener);

参数 function listener

监听到事件时的回调函数

参数 Object res

属性	类型	说明
usbmount	String	Usb 是否挂载
mountpath	String	Usb 挂载路径

示例代码：

```
coocaaosapi.addUSBChangedListener(  
    function(res) {  
        console.log("USB_CHANGGED received! ismount: " + res.usbmount);  
        console.log("USB_CHANGGED received! mountpath: " + res.mountpath);}  
)
```

## coocaaosapi.removeUSBChangedListener(function callback)

移除 USB 状态事件监听【u 盘】

```
brocaster.removeEventListener("USB_CHANGGED", listener);
```

参数 function callback

监听到事件时的回调函数

## coocaaosapi.startBlueToothSetting()

启动蓝牙设置。

```
startapp.start(["action", "android.settings.BLUETOOTH_SETTINGS"], success, error);
```

## coocaaosapi.startLocalMedia()

启动本地媒体。

```
startapp.start(["com.tianci.localmedia", "com.tianci.localmedia.MainActivity"], success, error);
```

## coocaaosapi.startMessageBox()

启动消息盒子。

```
startapp.start(["action", "com.coocaa.action.MESSAGEBOX"], success, error);
```

## coocaaosapi.startSourceList()

启动信号源。

```
exec(success, error, 'CoocaaOSApi', 'launchSourceList', []);
```

## coocaaosapi.startSystemUpgrade(Object object)

启动升级界面。

```
startapp.start(["action", "android.settings.SYSTEM_UPGRADE"], success, error);
```

## coocaaosapi.startTVSetting(Object object)

启动电视设置。

```
startapp.start(["com.tianci.setting", "com.tianci.setting.TianciSetting"], success, error);
```

## coocaaosapi.startNetSetting(Object object)

启动网络设置。

```
startapp.start(["action", "android.settings.NETWORK_OPERATOR_SETTINGS"], success, error);
```

## coocaaosapi.startCommonNormalAction(Object object)

启动传参 action 页面。详见 startapp.start()方法说明

```
startapp.start([param1,param2,param3,param4,param5],str, success, error);
```

## coocaaosapi.startHomeTab(Object object)

跳转到指定 tab 上，intent 附带参数，key 是 jumpToPage，值是 tab 的 id，由运营提供。

```
startapp.start(["action", actionName],[{ 'jumpToPage': tabid }], success, error);
```

- 1、如果系统是 6.x 的，action 是 coocaa.intent.action.HOME
- 2、如果系统是 5.x 的，需要判断一下 persist.service.homepage.pkg 这个 prop，如果值是 com.tianci.movieplatform，那么 action 是 coocaa.intent.action.HOME.Translucent,否则 action 就是 coocaa.intent.action.movie.home

## coocaaosapi.startHomeCommonList(Object object)

启动主页二级列表页

```
startapp.start(["action","coocaa.intent.action.HOME_COMMON_LIST"],[{id:id}], success,error);
```

参数

属性	类型	默认值	必填	说明
id	String		是	由运营提供

## coocaaosapi.startByPackName(Object object)

包名方式启动对应 app

```
startapp.start(pkgname, success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
Pkgname	String		是	

## 2.2 网络接口

### coocaaosapi.isNetConnected(Object object)

获取网络连接状态。

```
exec(success, error, 'CoocaaOSApi', 'isNetConnected', []);
```

success 回调函数

参数 Object res

属性	类型	说明
isConnected	String	“true”/“false”：当前是否有网络连接

## coocaaosapi.getNetType(Object object)

获取网络类型。

```
exec(success, error, 'CoocaaOSApi', 'getNetType', []);
```

success 回调函数

参数 Object res

属性	类型	说明
networkType	string	网络类型（wifi/cable/...）

## coocaaosapi.getIpInfo(Object object)

获取网络参数。

```
exec(success, error, 'CoocaaOSApi', 'getIpInfo', []);
```

success 回调函数

参数 Object res

属性	类型	说明
ip	string	
Mac	String	
Gateway	String	
Netmask	String	
Dns0	String	
Dns1	String	

示例代码：

返回值：

```
{
  "dns1": "172.20.135.2",
  "mac": "60:42:7f:e1:31:52",
  "gateway": "172.20.146.1",
  "netmask": "255.255.255.0",
  "dns0": "172.20.135.1",
  "ip": "172.20.146.147"
}
```

## coocaaosapi.addNetChangedListener(function callback)

网络状态变更事件监听

brocaster.addListener("NET\_CHANGGED", listener);

参数 function listener

监听到事件时的回调函数

参数 Object res

属性	类型	说明
isConnected	boolean	当前是否有网络连接
networkType	string	网络类型（wifi/cable/...）

示例代码：

```
coocaaosapi.addNetChangedListener(  
  function(res) {  
    console.log(res.isConnected);  
    console.log(res.networkType);  
  }  
)
```

## coocaaosapi.removeNetChangedListener(function callback)

移除网络状态变更事件监听

brocaster.removeListener("NET\_CHANGGED", listener);

参数 function listener

监听到事件时的回调函数

## 2.3 下载接口

### coocaaosapi.createDownloadTask(Object object)

直接下载任务。

```
exec(success, error, 'CoocaaOSApi', 'createDownloadTask',  
  [{ 'url': downloadurl }, { 'md5': md5 }, { 'title': title }, { 'pkg': packageName },  
  { 'appid': appId }, { 'icon': iconUrl }]);
```

参数

属性	类型	默认值	必填	说明
downloadurl	String	-	是	
md5	string	-	是	
Title	string	-	是	
packageName	string	-	是	
appId	string	-	是	
iconUrl	String		是	

# coocaaosapi.addAppTaskListener()

APK 下载事件监听  
brocaster.addEventListener("APP\_TASK\_CALLBACK", listener);

参数 function callback  
监听到事件时的回调函数

参数 Object res

属性	类型	说明
cancelBubble	boolean	
returnValue	boolean	
srcElement	Object	
defaultPrevented	boolean	
timeStamp	number	
cancelable	boolean	
bubbles	boolean	
eventPhase	number	
currentTarget	Object	
target	Object	
type	string	
createtime	number	
progress	number	
code	number	
Status	String	
url	String	
name	string	
taskid	number	

示例代码：

```
coocaaosapi.addAppTaskListener(  
  function(res) {  
    console.log(res.taskid);  
    console.log(res.name);}  
)  
回调函数参数如下：  
{  
  "cancelBubble": false,  
  "returnValue": true,  
  "srcElement": null,  
  "defaultPrevented": false,  
  "timeStamp": 1557905734705,  
  "cancelable": false,  
  "bubbles": false,  
  "eventPhase": 0,
```

```
    "currentTarget": null,
    "target": null,
    "type": "APP_TASK_CALLBACK",
    "createtime": 1557905734243,
    "progress": 0,
    "code": 0,
    "status": "ON_STARTING",
    "url": "https://qd.myapp.com/myapp/qqteam/AndroidQQ/mobileqq_android.apk",
    "name": "qq 移动版",
    "taskid": 10
}
```

## coocaaosapi.removeAppTaskListener(function callback)

移除 APK 下载事件监听

```
brocaster.removeEventListener("APP_TASK_CALLBACK", listener);
```

参数 function callback

监听到事件时的回调函数

## coocaaosapi.pauseDownloadTask(Object object)

暂停下载接口。

```
exec(success, error, 'CoocaaOSApi', 'pauseDownloadTask', [{taskid: taskid}]);
```

参数 Object object

属性	类型	默认值	必填	说明
Taskid	String		是	

## coocaaosapi.resumeDownloadTask(Object object)

恢复下载接口。

```
exec(success, error, 'CoocaaOSApi', 'resumeDownloadTask', [{taskid: taskid}]);
```

参数 Object object

属性	类型	默认值	必填	说明
Taskid	String		是	

## coocaaosapi.deleteDownloadTask(Object object)

删除下载接口。

```
exec(success, error, 'CoocaaOSApi', 'deleteDownloadTask', [{taskid: taskid}]);
```

参数 Object object

属性	类型	默认值	必填	说明
----	----	-----	----	----

Taskid	String		是	
--------	--------	--	---	--

## 2.4 多媒体接口

### coocaaosapi.startCommonWebview(Object object)

启动 web 播放器。

```
startapp.start(["action", "app_browser.intent.action.PLAYER", "com.coocaa.app_browser"],
    [{ "extra.id": id }, { "extra.uri": uri }, { "extra.tips": tips }, { "extra.height": height },
    { "extra.width": width }, { "extra.http_call_url": call_url }, { "extra.type": type },
    { "extra.name": name }]],
    success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
id	String	-	否	
uri	string	-	是	
name	string	-	否	
Height	string	-	否	
width	string	-	否	
Call_url	String		否	
type	String		否	
name	String		否	

### coocaaosapi.addCommonListener(function callback)

web 播放器通用事件监听【新浏览器】

```
brocaster.addEventListener("COMMON_CHANGED", listener);
```

参数 function listener

监听到事件时的回调函数

参数 Object res

属性	类型	说明
web_player_event	String	网络播放器播放状态

res.web\_player\_event 的合法值

值	说明
on_start	网络播放器开始播放
on_complete	网络播放器播放完成
on_interrupt	网络播放器播放中断（按返回键等）
on_error	网络播放器播放失败

示例代码：

```
coocaaosapi.addCommonListener(
```



```
function(res) {
  if(res.web_player_event == "on_start") {
    //some codes...
  }else if(message.web_player_event == "on_complete") {
    //some codes...
  }else if(message.web_player_event == "on_interrupt") {
    //some codes...
  }
}
}
```

## coocaaosapi.removeCommonListener(function callback)

取消 web 播放器通用事件监听【新浏览器】

brocaster.removeEventListener("COMMON\_CHANGED", listener);

参数 function listener

监听到事件时的回调函数

## coocaaosapi.playOnlineMovie(Object object)

启动系统播放器。

exec(success, error, 'CoocaaOSApi', 'startOnLinePlayer',  
[{url: url}, {'name': name}, {'needparse': needparse},{'urlType':'url'}]);

参数 Object object

属性	类型	默认值	必填	说明
uri	string	-	是	
name	string	-	否	
needparse	boolean	False	否	

## 2.5 设备信息

### coocaaosapi.getBaseInfo()

获取本机磁盘和内存信息。

exec(success, error, 'CoocaaOSApi', 'getSpaceInfo', []);

success 回调函数

参数 Object res

属性	类型	说明
freeSpace	number	可用硬盘空间
totalSpace	number	总共硬盘空间

totalMem	number	总共内存大小
leftMem	number	可用内存大小

示例代码：

```
coocaaosapi.getBaseInfo(  
    success(res) {  
        console.log(res.freeSpace);  
        console.log(res.totalSpace);  
        console.log(res.totalMem);  
        console.log(res.leftMem);  
    }  
)
```

返回值如下：

```
{  
    "freeSpace": 630980608,  
    "totalMem": 484253696,  
    "totalSpace": 2039562240,  
    "leftMem": 111505408  
}
```

## coocaaosapi.getDeviceInfo(Object object)

获取设备信息(屏幕大小/机芯机型/激活 ID 等)。

exec(success, error, 'CoocaaOSApi', 'getDeviceInfo', []);

success 回调函数

参数 Object res

属性	类型	说明
panel	String	本机屏幕尺寸
chip	String	本机机芯型号
chipid	String	本机主芯片型号
sid	String	
model	String	本机机型
devid	String	设备 ID
emmcid	String	EMMC ID
activeid	String	激活 ID
brand	String	品牌
barcode	String	二维码
mac	String	MAC 地址
androidsdk	number	Android SDK 版本
version	String	本机软件系统版本

示例代码：

```
coocaaosapi.getDeviceInfo(  
    success(res) {  
        console.log(res.panel);  
        console.log(res.chip);  
        console.log(res.chipid);  
        console.log(res.sid);  
        console.log(res.model);  
        console.log(res.devid);  
        console.log(res.emmcid);  
        console.log(res.activeid);  
        console.log(res.brand);  
        console.log(res.barcode);  
        console.log(res.mac);  
        console.log(res.androidsdk);  
        console.log(res.version);  
    }  
)
```

```
success(res) {
    console.log(res.panel);
    console.log(res.chip);
    console.log(res.activeid);
}
)
```

返回值如下：

```
{
    "panel": "32",
    "chip": "8S47",
    "chipid": "MST-6A338",
    "sid": "ebc9f795-788e-4c8f-b0cc-cd5371e112f3",
    "model": "E2A",
    "devid": "c86ae6c2352dd8b8d6624c6af6d8a5ad",
    "emmcid": "11010030303447363002129d2fa2a400",
    "activeid": "31140974",
    "brand": "Skyworth",
    "barcode": "32E2AXX-S093662-Z180516-406D9",
    "mac": "2835452aa239",
    "androidsdk": 19,
    "version": "7.00.190222"
}
```

## coocaaosapi.getLocation(Object object)

获取设备位置信息。

exec(success, error, 'CoocaaOSApi', 'getLocation', []);

success 回调函数

参数 Object res

属性	类型	说明
location	String	地址

示例代码：

```
coocaaosapi.getBaseInfo(
    success(res) {
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名
        console.log(res.location);
    }
)
```

返回值如下：

```
{
    "location": "广东省,深圳市\","
```

```
}
```

## coocaaosapi.getCurTheme()

获取主题信息。

```
exec(success, error, 'CoocaaOSApi', 'getCurTheme', []);
```

success 回调函数

参数 Object res

属性	类型	说明
theme	String	地址

示例代码：

```
coocaaosapi.getCurTheme(  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    console.log(res.theme);  
  }  
)
```

## coocaaosapi.hasApk()

检测是否装有 apk。

```
startapp.check(pkg, success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
Pkgname	String		是	Apk 包名

success 回调函数

参数 Object res

属性	类型	说明
Status	Boolean	存在/不存在

示例代码：

```
coocaaosapi.hasApk(  
  Pkgname: 'com.coocaa.mall',  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    Console.log(res.status);  
  }  
)
```

## 2.6 用户信息

### coocaaosapi.startUserSettingAndFinish()

启动用户设置，登录成功就消失。

```
startapp.start(["action", "android.settings.ADD_ACCOUNT_SETTINGS"],  
               [{ 'needFinish': true }], success, error);
```

### coocaaosapi.startWeixinOrQQ(Object object)

账户 4.2 版本开始支持微信、qq 或二选一登录：启动用户设置，登录成功就消失。

```
startapp.start(["action", "android.settings.ADD_ACCOUNT_SETTINGS"],  
               [{ 'needFinish': true }, { 'type': type }], success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
needFinish	Boolean		是	启动用户设置，登录成功就消失
Type	String		是	Qq、微信或二选一

### coocaaosapi.startWeixinOrQQ2(Object object)

包名类名方式启动【账户版本 4.3 以上，酷开版本 5.5 以下】。

```
startapp.start(["com.tianci.user", "com.tianci.webview.AccountWebActivity"],  
               [{ 'needFinish': true }, { 'type': type }], success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
needFinish	Boolean		是	启动用户设置，登录成功就消失
Type	String		是	Qq、微信或二选一

### coocaaosapi.hasCoocaaUserLogin()

用户是否登录。

```
exec(success, error, 'CoocaaOSApi', 'hasCoocaaUserLogin', []);
```

success 回调函数

参数 Object res

属性	类型	说明
haslogin	Boolean	是否已登录状态

示例代码：

```
coocaaosapi.hasCoocaaUserLogin(  
  success(res) {  
    console.log(res.haslogin);  
  }  
);
```

```
}  
)
```

## coocaaosapi.getUserInfo()

获取用户信息。

```
exec(success, error, 'CoocaaOSApi', 'getUserInfo', []);
```

success 回调函数

参数 Object res

属性	类型	说明
sky_id	String	
update_image_times	String	
open_id	String	
visit_num	String	
loginFlag	String	
external_info	Array.<object>	
Avatar	String	
tel1	String	
nick_name	String	
cardNo	String	
Balance	String	
create_date	String	
Email	String	
last_name	String	
Current	String	
Gender	String	
Avatars	Object	
third_account	String	
Qq	String	
Revenue	String	
Mobile	String	
Account_type	String	

object.external\_info 参数

属性	类型	说明
external_flag	String	qq/weixin
Login	Boolean	是否登录
Vusession	String	
Unionid	String	
bind_time	Number	
external_nickname	String	
external_id	String	
Vuserid	String	

external_avatar	String	
-----------------	--------	--

object.avatars 参数

属性	类型	说明
f_50	String	
f_70	String	
f_200	String	
f_250	String	
f_370	String	
f_500	String	
f_800	String	

示例代码：

```
coocaaosapi.getUserInfo(  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名  
    console.log(res.open_id);  
    console.log(res.nick_name);  
  }  
)  
)  
返回值如下 (demo) :  
  
  {  
    "sky_id": "200000000474",  
    "update_image_times": "0",  
    "open_id": "8151266f4ede11e6987500505687790a",  
    "visit_num": "566",  
    "loginFlag": "",  
    "external_info":  
    "[{"external_avatar\\":\\"\\",\\"vuserid\\":\\"\\",\\"external_id\\":\\"5E15FF5348E0B462797E4F4965B640C3\\",\\"external_nickname\\":\\"\\",\\"bind_time\\":1469062720000,\\"vusession\\":\\"\\",\\"login\\":false,\\"external_flag\\":\\"qq\\"},{\\"external_avatar\\":\\"http://thirdwx.qlogo.cn/mmpopen/vi_32/DYAI0gq83erkBY5Jun4VqFRlXgcNXfJ0CFWIUfBIkdV6b7CcttYy8zswfgWzRJewDricGNb07yjmx9kcwiaIZlWA/132\\",\\"vuserid\\":\\"530287424\\",\\"external_id\\":\\"o-G_Utw3KfMLFKe6sUcaVoIx9mPY\\",\\"external_nickname\\":\\"北纬 36 度\\",\\"bind_time\\":1518143947000,\\"unionid\\":\\"o6sbmwL5YwqJLZs2318xFcRbagWg\\",\\"vusession\\":\\"9b71d60e144de9ece3d6\\",\\"login\\":false,\\"external_flag\\":\\"weixin\\"}]",  
    "avatar":  
    "http://thirdwx.qlogo.cn/mmpopen/vi_32/DYAI0gq83erkBY5Jun4VqFRlXgcNXfJ0CFWIUfBIkdV6b7CcttYy8zswfgWzRJewDricGNb07yjmx9kcwiaIZlWA/132",  
    "tel1": "",  
    "nick_name": "          1",  
    "cardNo": "1101415325916",  
    "balance": "0",  
    "create_date": "1469062824000",  
    "email": "",  
    "last_name": "          ",
```

```

        "current": "true",
        "gender": "1",
        "avatars":
    "{ \"f_70\": \"http://thirdwx.qlogo.cn/mmopen/vi_32/DYAI0gq83erkBY5Jun4VqFRlXgcNXfJ0CFWIUfBIkdV6b7CcttYy8zswfgWzRJewDricGNb07yjmx9kcwiaIZlWA/132\", \"f_50\": \"http://thirdwx.qlogo.cn/mmopen/vi_32/DYAI0gq83erkBY5Jun4VqFRlXgcNXfJ0CFWIUfBIkdV6b7CcttYy8zswfgWzRJewDricGNb07yjmx9kcwiaIZlWA/132\", \"f_800\": \"http://thirdwx.qlogo.cn/mmopen/vi_32/DYAI0gq83erkBY5Jun4VqFRlXgcNXfJ0CFWIUfBIkdV6b7CcttYy8zswfgWzRJewDricGNb07yjmx9kcwiaIZlWA/132\", \"f_370\": \"http://thirdwx.qlogo.cn/mmopen/vi_32/DYAI0gq83erkBY5Jun4VqFRlXgcNXfJ0CFWIUfBIkdV6b7CcttYy8zswfgWzRJewDricGNb07yjmx9kcwiaIZlWA/132\", \"f_500\": \"http://thirdwx.qlogo.cn/mmopen/vi_32/DYAI0gq83erkBY5Jun4VqFRlXgcNXfJ0CFWIUfBIkdV6b7CcttYy8zswfgWzRJewDricGNb07yjmx9kcwiaIZlWA/132\", \"f_250\": \"http://thirdwx.qlogo.cn/mmopen/vi_32/DYAI0gq83erkBY5Jun4VqFRlXgcNXfJ0CFWIUfBIkdV6b7CcttYy8zswfgWzRJewDricGNb07yjmx9kcwiaIZlWA/132\", \"f_200\": \"http://thirdwx.qlogo.cn/mmopen/vi_32/DYAI0gq83erkBY5Jun4VqFRlXgcNXfJ0CFWIUfBIkdV6b7CcttYy8zswfgWzRJewDricGNb07yjmx9kcwiaIZlWA/132\" }",
        "third_account": "1",
        "qq": "",
        "revenue": "0",
        "mobile": "13728755689",
        "account_type": "1"
    }

```

## coocaaosapi.getUserAccessToken()

获取用户 access\_token。

```
exec(success, error, 'CoocaaOSApi', 'getUserAccessToken', []);
```

success 回调函数

参数 Object res

属性	类型	说明
accesstoken	String	

示例代码：

```

coocaaosapi.getUserAccessToken(
    success(res) {
        console.log(res.accesstoken); //类似 2.b290780935a64bb0aed89fe8f509cxxx 的值
    }
)

```

## coocaaosapi.setCoocaaUserLogout(Object object)

用户退出登录。

```
exec(success, error, 'CoocaaOSApi', 'setCoocaaUserLogout', []);
```



### coocaaosapi.addUserChanggedListener()

账户状态变更事件监听

brocaster.addEventListener("USER\_CHANGGED", listener);

参数 function listener

监听到事件时的回调函数

参数 Object res

属性	类型	说明
haslogin	boolean	

示例代码:

```
coocaaosapi.addUserChanggedListener(  
  function(res) {  
    console.log(res.haslogin);  
  }  
)
```

### coocaaosapi.removeUserChanggedListener()

移除账户状态变更事件监听

brocaster.removeEventListener("USER\_CHANGGED", listener);

参数 function listener

监听到事件时的回调函数

## 2.7 日志接口

### coocaaosapi.notifyJSLogInfo(Object object)

Web 页面日志提交到酷开大数据服务器。

exec(success, error, 'CoocaaOSApi', 'notifyJSLogInfo', [{ 'eventId': eventId }, { 'params': ddata }]);

参数 Object object

属性	类型	默认值	必填	说明
eventId	String		是	自定义 Event ID
data	String		是	自定义字段

示例代码:

```
var _date = {  
  "page_name": "新手学习页面",  
  "user_openid": "12345",
```

```

}
_date = JSON.stringify(_date);
coocaaosapi.notifyJSLogInfo(
    eventId: "page_show",
    Data: _data,
    success(res) {
        console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名
    }
)

```

## coocaaosapi.notifyJSMessage(Object object)

Web 页面消息传递给酷开系统。

`exec(success, error, 'CoocaaOSApi', 'notifyJSMessage', [{webInfo: mywebinfo}]);`

参数 mywebinfo

属性	类型	默认值	必填	说明
mywebinfo	String		是	自定义字段

示例代码:

```

coocaaosapi.notifyJSMessage(
    Mywebinfo,
    success(res) {
        console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名
    }
)

```

## 2.8 支付接口

### coocaaosapi.purchaseOrder(Object object)

启动支付页面。

`exec(success, error, 'CoocaaOSApi', 'purchaseOrder',`  
`{`  
`{'appcode': appcode},`  
`{'tradeid': tradeid},`  
`{'productname': productname},`  
`{'productsubname': productsubname},`  
`{'producttype': producttype},`  
`{'specialtype': specialtype},`  
`{'amount': amount},`  
`{'count': count},`  
`{'imgurl': imgurl},`  
`{'spec': spec}}`  
`);`

);

#### 参数

属性	类型	默认值	必填	说明
appcode	String			
tradeid	string			
productname	string			
productsubname	string			
producttype	string			
specialtype	Object			
amount	Number			
count	number			
Imgurl	string		否	
Spec	String		否	
success	function			
fail	function			
complete	function			

#### 示例代码：

```
coocaaosapi.purchaseOrder(  
  Appcode:'1001',  
  Tradeid: '1000001',  
  productName: '包月',  
  Productsubname:'product detail',  
  producttype:'虚拟',  
  Specialtype: {'notify_url':'http://42.121.113.121:8090/aqiyiOrder/viewMain.html' },  
  Amount: 0.01,  
  Count: 0,  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名  
  }  
)
```

#### coocaaosapi.purchaseOrder2(Object object)

启动集成到 webSDK 内部的支付页面-----即 2.2.3 (含)"versionCode">=2020003 以上使用。

```
exec(success, error, 'CoocaaOSApi', 'purchaseOrder',  
  [{ 'appcode': appcode },  
  { 'Tradeid': Tradeid },  
  { 'ProductName': ProductName },  
  { 'SpecialType': SpecialType },  
  { 'amount': amount },  
  { 'ProductType': ProductType },  
  { 'payAction': payAction },
```

```
        { 'cmd': cmd },
        { 'token': token },
        { 'tel': tel }}
    );
```

参数

属性	类型	默认值	必填	说明
appcode	String			
tradeid	string			
productname	string			
specialtype	string			
amount	Number			
producttype	string			
payAction	string			
cmd	string			
token	String			
tel	String			
success	function			
fail	function			
complete	function			

示例代码：

```
coocaaosapi.purchaseOrder2({
    Appcode: '1001',
    Tradeid: '1000001',
    productName: '包月',
    producttype: '虚拟',
    Specialtype:
        JSON.stringify({ 'notify_url': 'http://42.121.113.121:8090/aqiyiOrder/viewMain.html' }),
    Amount: 0.01,
    success(res) {
        console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名
    }
})
```

## coocaaosapi.addPurchaseOrderListener()

支付状态事件监听

```
brocaster.addEventListener("PURCHASE_CALLBACK", listener);
```

参数 function listener

监听到事件时的回调函数

示例代码：

```
coocaaosapi.addPurchaseOrderListener(  
    function(res) {  
        console.log(res);  
    }  
)
```

## coocaaosapi.removePurchaseOrderListener(function callback)

取消支付状态事件监听

```
brocaster.removeEventListener("PURCHASE_CALLBACK", listener);
```

参数 function listener

监听到事件时的回调函数

## 2.9 优惠券接口

### coocaaosapi.startAllCoupon(Object object)

启动优惠券列表页。

```
startapp.start(["action", "coocaa.intent.action.ALLCOUPON"],  
    [{ "sign": sign }, { "openId": openId }, { "appId": appId },  
    { "businessLine": businessLine }, { "business_type": businessType }],  
    success, error);
```

参数

属性	类型	默认值	必填	说明
sign	string			
openId	string			
appId	string			
businessLine	string			
businessType	string			

businessLine 的合法值

值	说明
Movie	影视
Edu	教育
...	其它待续

businessType 的合法值

值	说明
0	影视
1	教育
2	其它待续

示例代码：

```
coocaaosapi.startAllCoupon(  
  Sign: 'fs2dfa2awe3io2ljvwcnm08dbe476g9eyushcoocaa',  
  openId: 'a0742646762811e5ba4200163e022eda',  
  appId: 'CC0001',  
  businessLine: 'movie',  
  businessType: '0',  
  success(res) {  
    console.log(res); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
  }  
)
```

coocaaosapi.startMyCoupon(Object object)

启动我的优惠券。

```
startapp.start(["action", "coocaa.intent.action.MYCOUPON"],  
  [{ "sign": sign }, { "openId": openId }, { "appId": appId },  
  { "businessLine": businessLine }, { "business_type": businessType }],  
  success, error);
```

参数

属性	类型	默认值	必填	说明
sign	string			
openId	string			
appId	string			
businessLine	string			
businessType	string			

businessLine 的合法值

值	说明
Movie	影视
Edu	教育
...	其它待续

businessType 的合法值

值	说明
0	影视
1	教育
2	其它待续

示例代码：

```
coocaaosapi.startMyCoupon(  
  Sign: 'fs2dfa2awe3io2ljvwcnm08dbe476g9eyushcoocaa',  
  openId: 'a0742646762811e5ba4200163e022eda',  
  appId: 'CC0001',  
  businessLine: 'movie',  
  businessType: '0',  
  success(res) {
```

```
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名
    }
)
```

## 2.10 广告接口

### coocaaosapi.getAdData(Object object)

获取广告数据【图文广告传 appid+game\_id,game\_scene,game\_panel,game\_position】【视频广告传 appid+activity\_id,task\_id,】。

```
exec(success,error,'CoocaaOSApi','callAdBusiness',
    [{action:'getAdData'},
    {'appid':appid,'params':{'game_id':game_id,'game_scene':game_scene,'game_panel':
    game_panel,'game_position':game_position,'activity_id':activity_id,'task_id':task_id}}]
);
```

参数

属性	类型	默认值	必填	说明
appid	String		是	
game_id	string		*	
game_scene	string		*	
game_panel	string		*	
game_position	string		*	
activity_id	string		*	
task_id	string		*	
success	function		否	
fail	function		否	
complete	function		否	

success 回调函数

参数 Object res

属性	类型	说明
adMsg	Object	

res.adMsg 参数

属性	类型	说明
total	Number	返回的广告数据条数
schedules	Array.<object>	广告具体信息
其它参数		比较多，请参考以下返回值 demo，具体实现时请确认是否都需要

示例代码：

```
coocaaosapi.getAdData(
    appid:'CCADTV10015',
```

```

activity_id: '117',
task_id: '15' ,
success(res) {
    console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名
    var ADMsg= JSON.parse(res.adMsg);
    if(ADMsg.total > 0){
        //广告曝光
        sentThirdAdshow("video",ADMsg);
        sentThirdAdshow("videoStart",ADMsg);
        var url = ADMsg.schedules[0].content;
        console.log("广告数据正常 url:"+url);
        //播放视频广告
        coocaaosapi.startCommonWebview("", url, "广告视频", "1080", "1920", "", "广告 1",
            "广告 2", function(message) {console.log(message);},
            function(error) {console.log("startCommonWebview-"+error); });
    }
}
)

```

返回数据（demo）：

```

{
    "ad_setting": {
        "client_req_timeout": "1500",
        "min_space": "20",
        "monitor_test": "",
        "monitor_type": "0",
        "power_off_ad": "0",
        "sdk_download_type": "10000000",
        "system_time": "1546825830",
        "view_req_timeout": "1500"
    },
    "client_ip": "172.20.139.132",
    "data_type": "json",
    "db_path": "",
    "general_track_url": "",
    "interval": 20917,
    "next_time": 1546846747,
    "package_md5": "",
    "schedules": [{
        "adspace_id": "CCADTV10015",
        "app_type_id": -1,
        "begin_time": 1546531200,
        "bootAd": false,
        "bootImage": false,
        "bootVideo": false,
    }
    ]
}

```



```
        "caption": "",
        "click_event": "",
        "click_sdk_tracks": [],
        "click_tracks": [],
        "content":
"http://beta-res.hoisin.coocaatv.com/video/20190103/20190103165135825481.ts",
        "currHourRes": true,
        "dataFromCache": false,
        "effect": "",
        "end_time": 1549295999,
        "extend_param": {},
        "height": 1080,
        "isAssertFile": false,
        "isLocalScreensaver": false,
        "media_md5": "64dbc6672cfb2d96d48dbdce7b68d333",
        "media_size": 3787260,
        "media_type": "video",
        "needSyncRes": false,
        "nextHourRes": true,
        "onlineAd": true,
        "order_id": "020181210000006",
        "play_end_sdk_tracks": [],
        "play_start_sdk_tracks": [],
        "player_end_tracks":
["https://data-hoisin.coocaa.com/track?mac=bcec23461b11&model=model&ip=127.0.0.1&provi
nce=51&city=518000&sid=2019010403&time=1540202820374&did=10001"],
        "player_start_tracks":
["https://data-hoisin.coocaa.com/track?mac=bcec23461b11&model=model&ip=127.0.0.1&provi
nce=51&city=518000&sid=2019010402&time=1540202820374&did=10001"],
        "position_x": 0,
        "position_y": 0,
        "relationInfo": {
            "content": [],
            "relationInfoContent": [],
            "type": ""
        },
        "relation_info": "{\"content\":[],\"type\":"\""}",
        "resSavePath":
"/data/data/com.coocaa.app_browser/files/cc_ad_sdk/20190103165135825481.ts",
        "resTempSavePath":
"/data/data/com.coocaa.app_browser/files/cc_ad_sdk/temp/20190103165135825481.ts",
        "scheduleStatus": "UNKNOWN",
        "schedule_id": "S20190104000329",
        "schedule_md5": "5a8c7ba2db20b42372187e8e7ba9eabd",
```

```

        "sdk_track": [],
        "show_time": 15,
        "subscript": {
            "show_time": 0,
            "text": "",
            "type": "",
            "vice_text": ""
        },
        "time_offset": 0,
        "track_url":
["https://data-hoisin.coocaa.com/track?mac=bcec23461b11&model=model&ip=127.0.0.1&provi
nce=51&city=518000&sid=2019010401&time=1540202820374&did=10001"],
        "videoPausedAd": false,
        "videoTimelineAd": false,
        "webpAnimate": false,
        "width": 1920
    }],
    "sys_tracker": "http://tv.cctracker.com/hoisin/",
    "total": 1
}

```

## coocaaosapi.submitAdData(Object object)

提交内部广告数据。

```

exec(success,error,'CoocaaOSApi','callAdBusiness',
    [{ 'action': 'submitCoocaaData' }, { 'baseinfo': ad_base_info, 'eventid': 'ad_show',
    'params': { 'game_id': game_id, 'game_scene': game_scene, 'game_panel': game_panel,
    'game_position': game_position, 'activity_id': activity_id, 'task_id': task_id, 'result': result } } ]
);

```

参数

属性	类型	默认值	必填	说明
ad_base_info	String		是	广告基本信息，参考 <code>getAdData</code> 返回值的 <code>schedule[0]</code> 对象的信息
game_id	string		*	
game_scene	string		*	
game_panel	string		*	
game_position	string		*	
activity_id	string		*	
task_id	string		*	
Result	String		*	'true':视频广告开始播放; 'false':视频广告播放失败;
success	function		否	

fail	function		否	
complete	function		否	

object.Type 的合法值

值	说明
graphicAds	图文广告传 appid+game_id,game_scene,game_panel,game_position
videoAds	视频广告传 appid+activity_id,task_id,

示例代码:

```
coocaaosapi.submitAdData({
  Type: 'videoAd',
  Ad_base_info:JSON.stringify(ADMsg.schedules[0]), //ADMsg 来自 getAdData 返回值
  activity_id: '117',
  task_id: '15',
  success(res) {
    console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名
  }
})
```

## coocaaosapi.submitThirdAdData()

提交第三方广告数据。

```
exec(success,error,'CoocaaOSApi','callAdBusiness',
    [{'action':'submitThirdData'},
    {'scheduleId':scheduleId,'orderId':orderId,'adSpaceld':adSpaceld, 'trackUrl':trackUrl}]
);
```

参数

属性	类型	默认值	必填	说明
trackUrl	String		是	
scheduleId	String		是	
orderId	string		是	
adSpaceld	string		是	

trackUrl 参数:

采集点	说明
Video	Type 为 video 时, url 值为: JSON.stringify(ADMsg.schedules[0].track_url)
videoStart	Type 为 videoStart 时, url 值为: JSON.stringify(ADMsg.schedules[0].player_start_tracks)
videoEnd	Type 为 videoEnd 时, url 值为: JSON.stringify(ADMsg.schedules[0].player_end_tracks)
其它值	图文广告如何处理? 待完善...

示例代码:

```
//广告第三方监测
function sentThirdAdshow(type,msg) {
  var thirdUrl = "";
  if(type == "video"){
```

```

        thirdUrl = JSON.stringify(msg.schedules[0].track_url);
    }
    else if(type == "videoStart"){
        thirdUrl = JSON.stringify(msg.schedules[0].player_start_tracks);
    }
    else if(type == "videoEnd"){
        thirdUrl = JSON.stringify(msg.schedules[0].player_end_tracks);
    }
    coocaaosapi.submitThirdAdData( thirdUrl,
        msg.schedules[0].schedule_id,
        msg.schedules[ 0].order_id,
        msg.schedules[0].adspace_id,
        function (msg) {console.log("submitThirdAdData success=="+msg);},
        function (err) {console.log("submitThirdAdData err=="+err);});
}

```

## 2.11 影视模块

### coocaaosapi.startMovieHome()

启动影视中心。

```
startapp.start(["action", "coocaa.intent.movie.home"], success, error);
```

### coocaaosapi.startMovieList()

启动影视列表页。

```
startapp.start(["action", "coocaa.intent.movie.list"],[{id: listid}], success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
listid	string		是	影视列表 id

示例代码：

```

coocaaosapi.startMovieList(
    listid: "1",
    success(res) {
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名
    }
)

```

### coocaaosapi.startMovieDetail()

启动影视详情页。

```
startapp.start(["action", "coocaa.intent.movie.detailinfo"],[{'id': detailid}]], success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
detailid	string		是	影视 id

示例代码：

```
coocaaosapi.startMovieDetail(  
    detailid: "1",  
    success(res) {  
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

### coocaaosapi.startVideospecial()

启动轮播专题[一级]。

```
startapp.start(["action", "coocaa.intent.movie.videospecial"],[{'topicCode': detailid}]],  
    success, error);
```

参数

属性	类型	默认值	必填	说明
detailid	string		是	影视 id

示例代码：

```
coocaaosapi.startVideospecial(  
    detailid: "1",  
    success(res) {  
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

### coocaaosapi.startVideospecial2()

启动轮播专题[二级]。

```
startapp.start(["action", "coocaa.intent.movie.videospecial"],[{'pTopicCode': detailid}]],  
    success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
detailid	string		是	影视 id

示例代码：

```
coocaaosapi.startVideospecial2(  
    detailid: "1",  
    success(res) {  
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

```
}  
)
```

## coocaaosapi.startMovieTopic()

启动影视专题页。

```
startapp.start(["action", "coocaa.intent.movie.special"],[{'id': topicid}]], success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
topicid	string		是	影视专题 id

示例代码：

```
coocaaosapi.startMovieTopic(  
  topicid: "1",  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
  }  
)
```

## coocaaosapi.startMovieHomeSpecialTopic()

启动主页专题。

```
startapp.start(["action", "coocaa.intent.action.HOME_SPECIAL_TOPIC"],[{'id': id }]],  
  success, error);
```

参数

属性	类型	默认值	必填	说明
id	string		是	影视专题 id

示例代码：

```
coocaaosapi.startMovieHomeSpecialTopic(  
  id: "1",  
  success(res) {  
    console.log(res);  
  }  
)
```

## coocaaosapi.startMovieMemberCenter(Object object)

启动影视会员中心。

```
startapp.start(["action", "coocaa.intent.vip.center"],[{'source_id':sourceid}]], success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
sourceid	string		是	影视主题 id

示例代码：

```
coocaaosapi.startMovieMemberCenter(  
  sourceid: "1",  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
  }  
)
```

### coocaaosapi.startMovieHistory(Object object)

启动影视历史。

```
startapp.start(["action", "coocaa.intent.movie.history"], success, error);
```

## 2. 12 商城模块

### coocaaosapi.startAppShop()

启动酷开商城首页。

```
startapp.start(["action", "coocaa.intent.action.MALL_HOME"], success, error);
```

### coocaaosapi.startAppShopList()

启动酷开商城列表页。

```
startapp.start(["action", "coocaa.intent.action.MALL_LIST"], [{ "id": id, "title": title }],  
  success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
Id	string		是	
Title	String		是	

示例代码：

```
coocaaosapi.startAppShopList(  
  Id: '111',  
  Title: 'test',  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
  }  
)
```

### coocaaosapi.startAppShopDetail()

启动购物图文详情页。

startapp.start(["action", "coocaa.intent.action.MALL\_DETAIL"], [{"id": id}], success, error);

参数 Object object

属性	类型	默认值	必填	说明
Id	string		是	

示例代码：

```
coocaaosapi.startAppShopDetail(  
    Id:'111',  
    success(res) {  
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

## coocaaosapi.startAppShopVideo(Object object)

启动酷开商城视频详情页。

startapp.start(["action", "coocaa.intent.action.MALL\_VIDEO"],  
 [{"id": id}, {"url": url}, {"name": name}],  
 success, error);

参数

属性	类型	默认值	必填	说明
Id	string		是	
Url	string		是	
Name	string		是	

示例代码：

```
coocaaosapi.startAppShopVideo(  
    Id:'111',  
    Url:'',  
    Name:'test',  
    success(res) {  
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

## coocaaosapi.startAppShopBUYING()

启动酷开商城购物活动列表页。

startapp.start(["action", "coocaa.intent.action.MALL\_BUYING"], [{"id": id}], success, error);

参数

属性	类型	默认值	必填	说明
Id	string		是	

示例代码：



```
coocaaosapi.startAppShopBUYING(
  Id:'111',
  success(res) {
    console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名
  }
)
```

## coocaaosapi.startAppShopZone(Object object)

启动酷开商城专题页。

startapp.start(["action", "coocaa.intent.action.MALL\_ZONE"], [{"id": id}], success, error);

参数

属性	类型	默认值	必填	说明
id	string		*	如果 id 字段有值，底层 action 为 coocaa.intent.action.MALL_ZONE

示例代码：

```
coocaaosapi.startAppShopZone(
  Id:'111',
  success(res) {
    console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名
  }
)
```

## coocaaosapi.startAppShopZone2(Object object)

启动酷开商城专题页。

startapp.start(["action", "coocaa.intent.action.MALL\_LIST\_ZONE"], [{"pageId": id}], success, error);

参数

属性	类型	默认值	必填	说明
pageId	String		*	

示例代码：

```
coocaaosapi.startAppShopZone2({
  pageId:'111',
  success(res) {
    console.log(res.errMsg); //调用成功时: "xxx:ok" , 其中 xxx 为调用的接口名
  }
})
```

## coocaaosapi.startAppShopZoneList()

启动酷开商城专题列表页。

```
startapp.start(["action", "coocaa.intent.action.MALL_ZONE_LIST"], success, error);
```

## 2.13 应用模块

### coocaaosapi.startAppStore(Object object)

启动酷开应用商城。

```
startapp.start(["action", "coocaa.intent.action.APP_STORE_HOME"], success, error);
```

### coocaaosapi.startAppStoreSort()

启动酷开应用商城分类页。

```
startapp.start(["action", "coocaa.intent.action.APP_STORE_SORT"],[{sortid: sortid}],
                success, error);
```

参数

属性	类型	默认值	必填	说明
sortid	string		是	

示例代码：

```
coocaaosapi.startAppStoreSort(
    sortid:'111',
    success(res) {
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名
    }
)
```

### coocaaosapi.startAppStoreList(Object object)

启动酷开应用商城列表页。

```
startapp.start(["action", "coocaa.intent.action.APP_STORE_LIST"],[{listId: listid}],
                success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
listId	string		是	

示例代码：

```
coocaaosapi.startAppStoreList(
    listId:'111',
    success(res) {
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名
    }
)
```

)

## coocaaosapi.startAppStoreBD()

启动酷开应用商城榜单页。

```
startapp.start(["action", "coocaa.intent.action.APP_STORE_RANKING"],[{'rankId': rankid}],  
               success, error);
```

参数

属性	类型	默认值	必填	说明
rankId	string		是	

示例代码：

```
coocaaosapi.startAppStoreBD(  
    rankId: '111',  
    success(res) {  
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

## coocaaosapi.startAppStoreDetail(Object object)

启动酷开应用商城详情页。

```
startapp.start(["action", "coocaa.intent.action.APP_STORE_DETAIL"],[{'id': idorpgk}],  
               success, error);
```

参数

属性	类型	默认值	必填	说明
Id	string		是	可以传递 pkg 或者 id

示例代码：

```
coocaaosapi.startAppStoreDetail(  
    Id: '111',  
    success(res) {  
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

## coocaaosapi.startAppStoreZone(Object object)

启动酷开应用商城专题页。

```
startapp.start(["action", "coocaa.intent.action.ZONEPAGE"],[{'id': zoneid}], success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
zoneId	string		是	

示例代码：

```
coocaaosapi.startAppStoreZone(  
    zoneId:'111',  
    success(res) {  
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

### coocaaosapi.startMyApps(Object object)

启动我的应用。

```
startapp.start(["action", "coocaa.intent.action.MYAPP_CHILD_MODEL"], success, error);
```

```
startapp.start(["action", "coocaa.intent.action.APP_STORE_MYAPPS"], success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
mode	string		是	child:儿童模式, 默认非儿童模式

示例代码：

```
coocaaosapi.startMyApps(  
    mode:'child',  
    success(res) {  
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

## 2. 14 游戏模块

### coocaaosapi.startGameCenter()

启动酷游吧。

```
startapp.start(["action", "coocaa.intent.action.GAME_CENTER_HOME"], success, error);
```

### coocaaosapi.startGameArsenal(Object object)

启动游戏军火库。

```
startapp.start(["action", "coocaa.intent.action.GAME_CENTER_ARSENAL"], success, error);
```

### coocaaosapi.startGameCenterDetail(Object object)

启动酷游吧游戏详情页。

```
startapp.start(["action", "coocaa.intent.action.GAME_CENTER_DETAIL"],[{id: idorpgk}],  
    success, error);
```

#### 参数

属性	类型	默认值	必填	说明
id	string		是	可以传递 pkg 或者 id

示例代码：

```
coocaaosapi.startGameCenterDetail(  
  id:'111',  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
  }  
)
```

### coocaaosapi.startGameCenterList(Object object)

启动酷游吧游戏列表页。

```
startapp.start(["action", "coocaa.intent.action.GAME_CENTER_LIST"],[{id: id,'title': title}],  
  success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
id	string		是	
Title	String		是	

示例代码：

```
coocaaosapi.startGameCenterList(  
  id:'111',  
  Title:'test',  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
  }  
)
```

### coocaaosapi.startGameCenterZone(Object object)

启动酷游吧游戏专题页。

```
startapp.start(["action", "coocaa.intent.action.GAME_CENTER_ZONE"],[{id: id}],  
  success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
Id	string		是	

示例代码：

```
coocaaosapi.startGameCenterZone(  
  Id:'111',  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
  }  
)
```

```
)
```

## coocaaosapi.startMyGames()

启动我的游戏。

```
startapp.start(["action", "coocaa.intent.action.GAME_CENTER_MYGAME"], success, error);
```

## 2.15 启动 Web 页接口

### coocaaosapi.startNewBrowser(Object object)

启动新版本浏览器。

```
startapp.start(["action", "coocaa.intent.action.browser", "com.coocaa.app_browser"],  
    [{ "url": url }], success, error);
```

启动新版本浏览器(背景不透明)。

```
startapp.start(["action", "coocaa.intent.action.browser.no_trans", "com.coocaa.app_browser"],  
    [{ "url": url }], success, error);
```

启动新版本浏览器(无路径，即不启动多层 activity)。

```
startapp.start(["action", "coocaa.intent.action.browser.no_route", "com.coocaa.app_browser"],  
    [{ "url": url }], success, error);
```

启动新版本浏览器(背景不透明，无路径)。

```
startapp.start(["action", "coocaa.intent.action.browser.no_trans.no_route",  
    "com.coocaa.app_browser"], [{ "url": url }], success, error);
```

启动新版本浏览器(跟随主题背景)。

```
startapp.start(["action", "coocaa.intent.action.browser.theme_bg", "com.coocaa.app_browser"],  
    [{ "url": url }], success, error);
```

参数 Object object

属性	类型	默认值	必填	说明
Url	String		是	Web 页地址

示例代码：

```
coocaaosapi.startNewBrowser(  
    Url: "https://www.baidu.com",  
    success(res) {  
        console.log(res); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名  
    }  
)
```

## 2.16 页面生命周期及事件

页面事件包括以下，主要是 `cordova.js` 里的客制化修改：

- 1. 页面生命周期事件注册
- 2. 遥控按键注册
- 3. 页面退出等

页面生命周期事件	说明
deviceready	Coocaaosapi 所需 natvie 环境全部准备完成
activated	?
Pause	web 页面进入后台时触发
Resume	Web 页面重新进入前台时触发
online	
offline	

遥控按键事件	说明
homebutton	主页键
backbutton	返回键
Backbuttondown	返回键按下时
Menubutton	菜单键

```
//let native know we need to override the back button.
exec(null, null, "CoreAndroid", "overrideBackbutton", [true/false]);

// let native know we need to override the back button.
exec(null, null, "CoreAndroid", "overrideHomebutton", [true/false]);

// generic button bind used for volumeup/volumedown buttons
exec(null, null, "CoreAndroid", "overrideButton", [buttonName, true/false]);
```

buttonName 合法值	说明
volumeup	
volumedown	
其它。。	

```
// Let native code know we are all done on the JS side.
// Native code will then un-hide the WebView.
exec(onMessageFromNative, null, "CoreAndroid", 'messageChannel', []);
onMessageFromNative 回调函数 res 对象参数说明：
```

字段	格式	说明
action	String	按键说明

```
function onMessageFromNative(msg) {
    var action = msg.action;
    switch (action) {
        // Button events
        case 'backbutton':
        case 'homebutton':
```

```

        case 'backbuttondown':
        case 'menubutton':
        case 'searchbutton':
        case 'pause':
        case 'volumedownbutton':
        case 'volumeupbutton':
        case 'resume':

```

```
exec(null, null, "CoreAndroid", "show", []);
```

## coocaaosapi.exitWeb(Object object)

退出当前 web 页。

```
exec(null, null, APP_PLUGIN_NAME, "exitApp", []);
```

退出当前所有打开的 web 页

```
exec(null, null, APP_PLUGIN_NAME, "exitAll", []);
```

## 2.17 安卓小程序接口

### coocaaosapi.startAppX2()

启动安卓小程序页面。

```
startapp.startservice(["appx.intent.service.launcher.Start","com.coocaa.app_browser",
    [uri, {'pre_load': true}]], success, error);}
```

参数 Object object

属性	类型	默认值	必填	说明
Url	String		是	安卓小程序 url 地址
preLoad	String		是	是否预加载

示例代码：

```

coocaaosapi.startAppX2(
    Url: "appx://com.coocaa.appx.member.guide",
    preLoad: "false",
    success(res) {
        console.log(res.errMsg); //调用成功时: "xxx:ok" ， 其中 xxx 为调用的接口名
    }
)

```

## 2.18 广播接口



coocaaosapi.addGlobalBroadcastListener()

添加全局广播监听。  
增加了监听全局广播的能力，只要页面还在，无论是否被覆盖，都能监听到全局广播。  
brocaster.addGlobalEventListener( action,listener);

参数 Object object

属性	类型	默认值	必填	说明
action	String		是	需要监听的广播消息
listener	function		是	接收到广播消息时的回调函数

示例代码：

```
coocaaosapi.addGlobalBroadcastListener(  
  action: "com.coocaa.appx.broadcasr.action.appx_launched",  
  callback(res) {  
    console.log('callback ok..' +JSON.stringify(res));  
  },  
  success(res) {  
    console.log(res.errMsg); //调用成功时: "xxx:ok" ，其中 xxx 为调用的接口名  
  }  
)
```

coocaaosapi.removeGlobalBroadcastListener(Object object)

移除全局广播监听  
brocaster.removeGlobalEventListener( action, listener);

coocaaosapi.sendBroadcast(Object object)

发送全局广播消息