华为开放PUSH平台富媒体接口文档

Huawei Technologies Co., Ltd.

华为技术有限公司

Table of Contents 目　录

1 PUSH 开放平台富媒体接口概述

1.1 系统概述

目的：为了丰富华为Push富媒体呈现能力，使业务做一些简单的富媒体Html/JavaScript开发即可使用Android基础能力，如GPS、加速计、应用拉起、播放音、视频。

1.2 典型应用交互流程

- 开发者根据华为 Push 提供的 JavaScript lib 进行简单的开发和测试
- 在联盟上新建富媒体推送，并复制对应 Html 与 javaScript 内容
- 推送到手机，并查看效果

1.3 注意事项

开发者本地开发调试的过程中，可以使用http://open.hicloud.com/android/push_destop1.0.js进行测试接口是否OK。实际发送富媒体的时候，建议去除引用该文件，节省客户端流量，该文件只适用于Pc开发使用。

2 富媒体 JavaScript 开发接口

2.1 播放音频 JavaScript 接口

1. prepare(successCallback, errorCallback, statusCallback, options)

说明：准备音频文件接口，该函数主要是进行音频文件校验，目前仅支持http/https、下载到本地音频（相对路径）。

参数		描述	是否必选
successCallback		音频文件校验成功回调，可在回调里面执行Play()	否
errorCallback		音频准备、播放过程失败等回调	否
statusCallback		获取当前播放位置/音频时长回调	否
options	url	音频地址。网络路径或者相对路径（下载打开的富媒体类型）	是
	pauseOnActivityPause	在当前Activity onPause时，是否暂停播放。默认true	否

2. play()

说明：播放音频

3. **pause()**

   说明：暂停播放

4. **stop()**

   说明：停止播放

5. **seekto(milliseconds)**

   说明：跳转到 milliseconds 时刻播放

6. **getPlayingStatus(frequently)**

   说明：定时返回当前的播放状态，包括播放位置/音频时长。回调statusCallback

接口示例

```
<html>
    <head>
        <title>AUDIO Example</title>
        <!—Pc broswer  测试javascript用，实际推送时可去除  -->
        <script type="text/javascript" charset="utf-8" src="http://open.hicloud.com/android/push_destop1.0.js"></script>
        <script type="text/javascript" charset="utf-8">
            function onDeviceReady(){
                console.log("when device ready,will print this log");
            }
            var url = null;
            var ready = false;
            var mediaTimer = null;
            //播放
            function start(src) {
                if (url == src && ready == true) {
                    audio.play();
                } else {
                    url = src;
                    audio.stop();
                    var options = {
                        "url" : url,
                        "pauseOnActivityPause" : true
                    }
                    audio.prepare(success, error, status, options);
                }
            };
            //音频准备成功回调函数
            function success(json) {
                console.log(JSON.stringify(json));
                ready = true;
                audio.play();
                //获取当前播放位置/时长
                audio.getPlayingStatus(1000);
                var log = document.getElementById('log');
                log.innerHTML += "\n" + startTime() + "\n" + JSON.stringify(json);
            };
            //错误回调函数
            function error(json) {
                alert(JSON.stringify(json));
                stop();
            };
            //播放状态回调函数
            function status(json) {
                //返回当前位置/时长
                console.log("on audio status :" + JSON.stringify(json));
                var log = document.getElementById('log');
                log.innerHTML += "\n" + startTime() + "\n" + JSON.stringify(json);
            };
            //从N毫秒开始播放
            function seekto() {
                audio.seekto(10000);
            }
            //暂停
            function pause() {
                audio.pause();
            };
            //停止
            function stop() {
                audio.stop();
                url = null;
                ready = false;
                if (mediaTimer != null) {
                    clearInterval(mediaTimer);
                }
                mediaTimer = null;
            };
            //获取时间
            function startTime() {
                var today = new Date()
                var h = today.getHours()
                var m = today.getMinutes()
                var s = today.getSeconds()
                // add a zero in front of numbers<10
```

```
                        m = checkTime(m)
                        s = checkTime(s)
                        return h + ":" + m + ":" + s;
                }
                function checkTime(i) {
                        if (i < 10) {
                                i = "0" + i
                        }
                        return i
                }
        </script>
</head>
<body style="width: 1000px;height: 1000px">
        <button onclick="javascript:start('http://xxxxxx/xxxxx1.mp3');">
                Play Music 1
        </button>
        <button onclick="javascript:start('http://xxxxxx/xxxxx2.mp3');">
                Play Music 2
        </button>
        <button onclick="javascript:seekto();">
                seekto 10s
        </button>
        <button onclick="pause()">
                pause
        </button>
        <button onclick="stop()">
                stop
        </button>
        <div id="log">
                &nbsp;
        </div>
</body>
</html>
```

2.2 播放视频 JavaScript 接口

1. play(successCallback, errorCallback, options)

说明：播放视频接口.

参数	描述	是否必选
successCallback	成功发出视频播放请求回调	否
errorCallback	发出视频播放请求之前失败回	否

		调	
options	url	视频播放地址，仅支持网络视频地址	是
	mime-type	视频类型，默认是"video/*"	否
	package-name	指定由哪个视频播放器播放	否

接口示例

```
<html>
    <head>
        <title>VIDEO Example</title>
         <!—Pc broswer  测试javascript用，实际推送时可去除  -->
        <script type="text/javascript" charset="utf-8" src="http://open.hicloud.com/android/push_destop1.0.js"></script>
        <script type="text/javascript" charset="utf-8">
            function onDeviceReady(){
                console.log("when device ready,will print this log");
            }
            //播放视频
            function play_video_1() {
                var options = {
                    "url":"http://xxxxx/xxxx.3gp",
                }
                video.play(onSuccess,onError,options);
            }
            //使用指定视频播放应用打开，示例：使用pps应用
            function play_video_2() {
                var options = {
                    "url":"http://xxxxx/xxxxxx.mp4",
                    "mime-type":"video/mp4",
                    "package-name":"tv.pps.mobile"
                }

                video.play(onSuccess,onError,options);
            }

            function onSuccess(json){
                console.log(JSON.stringify(json));
            }
            function onError(json){
                console.log(JSON.stringify(json));
                }

            </script>
        </head>
        <body>
            <button onclick="javascript:play_video_1();">
                play_video_1
            </button>
            <button onclick="javascript:play_video_2();">
                play_video_2
            </button>
        </body>
</html>
```

2.3 加速度（摇一摇）JavaScript 接口

1. start (successCallback, errorCallback, options)

说明：开启加速度服务

参数		描述	是否必选
successCallback		成功获取到加速度回调	否
errorCallback		获取加速度失败	否
options 触发条件	minAccuracy	最小精度	1、Low 2、Medium 3、High
	minAccelX	最小水平横向加速度	默认 10
	minAccelY	最小水平竖向加速度	默认 10
	minAccelZ	最小垂直加速度	默认 10

2. stop ()

说明：停止加速度服务

3. playSound (options)

说明：播放短促音频

参数	描述	是否必选

options 额外参数	soundType	音频类型	1、TYPE_SHAKE 目前仅支持1种

接口示例

```
<html>
    <head>
        <title>ACCEL   Example</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
         <!—Pc broswer 测试javascript用，实际推送时可去除 -->
        <script type="text/javascript" charset="utf-8" src="http://open.hicloud.com/android/push_destop1.0.js"></script>
        <script type="text/javascript" charset="utf-8">
            function onDeviceReady(){
                console.log("when device ready,will print this log");
            }

            var playSoundAfterSuccess = true;
            function start() {
                var options = {
                    "minAccuracy" : 2,
                    "minAccelX" : 14,
                    "minAccelY" : 14,
                    "minAccelZ" : 14
                };
                accelerometer.start(onSuccess, onError, options);
            }
            function stop() {
                accelerometer.stop();
            }
            function onSuccess(json) {
                var log = document.getElementById('log');
                log.innerHTML += "\n" + startTime() + "\n" + JSON.stringify(json);
                console.log(JSON.stringify(json));
                if (playSoundAfterSuccess) {
                    stop();
                    var options = {
                        "soundType" : "TYPE_SHAKE",
                    };
                    accelerometer.playSound(options);
                }
            }
            function onError(json) {
                var log = document.getElementById('log');
                log.innerHTML += "\n" + startTime() + "\n" + JSON.stringify(json);
                console.log(JSON.stringify(json));
                stop();
            }
            function startTime() {
                var today = new Date()
                var h = today.getHours()
                var m = today.getMinutes()
                var s = today.getSeconds()
                // add a zero in front of numbers<10
                m = checkTime(m)
                s = checkTime(s)
                return h + ":" + m + ":" + s;
            }
            function checkTime(i) {
                if (i < 10) {
                    i = "0" + i
                }
                return i
            }
        </script>
    </head>
    <body>
        <button onclick="javascript:start();">
            start_audio_shake
        </button>
        <button onclick="javascript:stop();">
            stop
        </button>
        <div id="log">
            &nbsp;
        </div>
    </body>
</html>
```

2.4 地理位置接口

1. getLocation (successCallback, errorCallback, options)

说明：获取地理位置信息

参数	描述	是否必选

successCallback		成功获取到地理位置	否
errorCallback		获取地址位置失败	否
options 触发条件	useGps	是否使用 GPS（建议定位时使用 GPS，网络定位依赖 google 框架服务，部分 ROM 可能不存在该服务）	否，默认 false
	keepLoc	是否监听多个位置	默认 true
	minTime	最小时间变化时返回位置信息	默认 0 ，实时获取
	minDistance	最小距离变化时返回位置信息	默认 0 ，实时获取

2. clearWatch ()

说明： 取消位置监听

接口示例

```
<html>
    <head>
        <title>GEO  Example</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <!—Pc broswer 测试javascript用，实际推送时可去除 -->
        <script type="text/javascript" charset="utf-8" src="http://open.hicloud.com/android/push_destop1.0.js"></script>
        <script type="text/javascript" charset="utf-8">
            function onDeviceReady(){
                console.log("when device ready,will print this log");
            }
            function getLocation_gps() {
                var options = {
                    "useGps" : true,
                    "keepLoc" : true,
                    "minTime" : 0,
                    "minDistance" : 0
                }
                geo.getLocation(onSuccess, onError, options);

            }

            function getLocation_network() {
```

```
            var options = {
                "useGps" : false,
                "keepLoc" : true,
                "minTime" : 0,
                "minDistance" : 0
            }
            geo.getLocation(onSuccess, onError, options);
        }

        function onSuccess(json) {
            var log = document.getElementById('log');
            log.innerHTML += "\n" + startTime() + "\n" + JSON.stringify(json);
            console.log(JSON.stringify(json));
        }

        function onError(json) {
            var log = document.getElementById('log');
            log.innerHTML += "\n" + startTime() + "\n" + JSON.stringify(json);
            console.log(JSON.stringify(json));
        }
        function startTime() {
            var today = new Date()
            var h = today.getHours()
            var m = today.getMinutes()
            var s = today.getSeconds()
            // add a zero in front of numbers<10
            m = checkTime(m)
            s = checkTime(s)

            return h + ":" + m + ":" + s;
        }
        function checkTime(i) {
            if (i < 10) {
                i = "0" + i
            }
            return i
        }
    </script>
</head>
<body>
    <button onclick="javascript:getLocation_network();">
        getLocation_network
    </button>
```

```
                <button onclick="javascript:getLocation_gps();">
                        getLocation_gps
                </button>
                <div id="log">
                        &nbsp;
                </div>
        </body>
</html>
```

2.5 打开应用接口

3. open (successCallback, errorCallback,statusCallback, options)

说明：打开指定应用

参数		描述	是否必选
successCallback		应用打开成功	否
errorCallback		应用打开失败	否
statusCallback		应用关闭后，返回OnActivityResult	否
options 条件	package-name	应用包名	是
	intent-uri	打开Activity的Intent URI字符串	否
	requestCode	匹配过滤OnActivityResult	否
	resultCode	匹配过滤OnActivityResult	否
	appmarket	应用不存在时是否打开应用市场	否（默认不打开）

接口示例

```
<html>
        <head>
                <title>APP   Example</title>
                <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
                 <!—Pc broswer  测试javascript用，实际推送时可去除  -->
                <script type="text/javascript" charset="utf-8" src="http://open.hicloud.com/android/push_destop1.0.js"></script>
                <script type="text/javascript" charset="utf-8">
```

```
function onDeviceReady(){
        console.log("when device ready,will print this log");
}
//openAPP_exist_with_result
function openAPP1() {
        var options = {
                "package-name" : "com.huawei.pushapptest",
                "intent-uri" :
"#Intent;component=com.huawei.pushapptest/.Activity2;S.key=hello%20%2Cthis%20is%20from%20intent%20extra;end",
                "requestCode" : 1,
                "resultCode" : -1
        }
        app.open(onSuccess, onError, onActivityResult, options);
}

//openAPP_exist_no_result
function openAPP2() {
        var options = {
                "package-name" : "com.huawei.pushapptest",
                "intent-uri" :
"#Intent;component=com.huawei.pushapptest/.Activity2;S.key=hello%20%2Cthis%20is%20from%20intent%20extra;end"
        }
        app.open(onSuccess, onError, null, options);
}

//openAPP_no_exist_appmarket_false
function openAPP3() {
        var options = {
                "package-name" : "com.ddshow"
        }
        app.open(onSuccess, onError, null, options);
}

// openAPP_no_exist_appmarket_true
function openAPP4() {
        var options = {
                "package-name" : "com.ddshow",
                "appmarket" : true
        }
        app.open(onSuccess, onError, null, options);
}

// openAPP_no_exist_no_appmarket
```

```
function openAPP5() {
    var options = {
        "package-name" : "com.ddshow",
        "appmarket" : true
    }
    app.open(onSuccess, onError, null, options);
}

function onSuccess(json) {
    var log = document.getElementById('log');
    log.innerHTML += "\n" + startTime() + "\n" + JSON.stringify(json);
    console.log(JSON.stringify(json));
}

function onError(json) {
    var log = document.getElementById('log');
    log.innerHTML += "\n" + startTime() + "\n" + JSON.stringify(json);
    console.log(JSON.stringify(json));
}

function onActivityResult(json) {
    var log = document.getElementById('log');
    log.innerHTML += "\n" + startTime() + "\n" + JSON.stringify(json);
    console.log(JSON.stringify(json));
}

function startTime() {
    var today = new Date()
    var h = today.getHours()
    var m = today.getMinutes()
    var s = today.getSeconds()
    // add a zero in front of numbers<10
    m = checkTime(m)
    s = checkTime(s)

    return h + ":" + m + ":" + s;
}

function checkTime(i) {
    if (i < 10) {
        i = "0" + i
    }
    return i
```

```
                    }

                </script>
            </head>
            <body>

                <button onclick="javascript:openAPP1();">
                    openAPP_exist_with_result
                </button>
                <button onclick="javascript:openAPP2();">
                    openAPP_exist_no_result
                </button>
                <button onclick="javascript:openAPP3();">
                    openAPP_no_exist_appmarket_false
                </button>
                <button onclick="javascript:openAPP4();">
                    openAPP_no_exist_appmarket_true
                </button>
                <button onclick="javascript:openAPP5();">
                    openAPP_no_exist_no_appmarket
                </button>
                <div id="log">
                    &nbsp;
                </div>
            </body>
        </html>
```