

# 支付宝容器jsAPI文档手册PDF版本

本版本为实验版本，为线下独立查看使用，受制于生成PDF程序的限制，样式问题还没有很好的解决，例如分页切图、代码高亮。  
推荐使用在线版本，方便实时查看jsAPI运行效果。  
如需搜索，使用阅读PDF软件自带功能即可。  
更多细节样式调整和目录索引探索建设中。  
生成时间: 2017-06-08 05:40:30

## V3.0.2 更新日志 2017-06-01

- 新增 API `ap.startBizService()` 启动某项支付宝业务服务。详情 (util/service/startBizService.html)
- 修复 API `ap.onResume()` 事件返回值 `data` 字段缺失的问题。详情 (navigation/event/onResume.html)
- 修复 运行环境对 Promise 的检测，不支持时仍然可以使用 `callback` 方式。
- 修复 Promise 方式调用 API 时需要等待 `ap.ready()` resolve 的问题。
- 修复 调用 `ap` 方法时，`ready` 事件延时可能导致的 `AlipayJSBridge` 对象未初始化的报错。
- 修复 事件移除接口 `ap.offXXX()` 不生效的问题。
- 修复 API `ap.onBLECharacteristicValueChange()` 事件注册不生效的问题。
- 修复 API `ap.onBLEConnectionStateChanged()` 事件注册不生效的问题。

## V3.0.0 更新日志 2017-05-19

- 新增 API `ap.openBluetoothAdapter()` 初始化本机蓝牙模块。详情 (device/bluetooth/openBluetoothAdapter.html)
- 新增 API `ap.closeBluetoothAdapter()` 关闭本机蓝牙模块。详情 (device/bluetooth/closeBluetoothAdapter.html)
- 新增 API `ap.getBluetoothAdapterState()` 获取本机蓝牙模块状态。详情 (device/bluetooth/getBluetoothAdapterState.html)
- 新增 API `ap.startBluetoothDevicesDiscovery()` 开始搜寻附近的蓝牙外围设备。详情 (device/bluetooth/startBluetoothDevicesDiscovery.html)
- 新增 API `ap.stopBluetoothDevicesDiscovery()` 停止搜寻附近的蓝牙外围设备。详情 (device/bluetooth/stopBluetoothDevicesDiscovery.html)
- 新增 API `ap.getBluetoothDevices()` 获取所有已发现的蓝牙设备。详情 (device/bluetooth/getBluetoothDevices.html)
- 新增 API `ap.getConnectedBluetoothDevices()` 根据 `service` 的 `uuid` 获取处于已连接状态的设备。详情 (device/bluetooth/getConnectedBluetoothDevices.html)
- 新增 API `ap.connectBLEDevice()` 连接低功耗蓝牙设备。详情 (device/bluetooth/connectBLEDevice.html)
- 新增 API `ap.disconnectBLEDevice()` 断开与低功耗蓝牙设备的连接。详情 (device/bluetooth/disconnectBLEDevice.html)

- 新增 API `ap.writeBLECharacteristicValue()` 向低功耗蓝牙设备特征值中写入数据。详情 (<device/bluetooth/writeBLECharacteristicValue.html>)
- 新增 API `ap.readBLECharacteristicValue()` 读取低功耗蓝牙设备特征值中的数据。详情 (<device/bluetooth/readBLECharacteristicValue.html>)
- 新增 API `ap.notifyBLECharacteristicValueChange()` 启用低功耗蓝牙设备特征值变化时的 `notify` 功能。详情 (<device/bluetooth/notifyBLECharacteristicValueChange.html>)
- 新增 API `ap.getBLEDeviceServices()` 获取蓝牙设备所有 `service`。详情 (<device/bluetooth/getBLEDeviceServices.html>)
- 新增 API `ap.getBLEDeviceCharacteristics()` 获取蓝牙设备所有 `characteristic`。详情 (<device/bluetooth/getBLEDeviceCharacteristics.html>)
- 新增 API `ap.onBluetoothDeviceFound()` 寻找到新的蓝牙设备时触发此事件。详情 (<device/bluetooth/onBluetoothDeviceFound.html>)
- 新增 API `ap.offBluetoothDeviceFound()` 移除寻找到新的蓝牙设备事件的监听。详情 (<device/bluetooth/offBluetoothDeviceFound.html>)
- 新增 API `ap.onBLECharacteristicValueChange()` 监听低功耗蓝牙设备的特征值变化事件。详情 (<device/bluetooth/onBLECharacteristicValueChange.html>)
- 新增 API `ap.offBLECharacteristicValueChange()` 移除低功耗蓝牙设备的特征值变化事件监听。详情 (<device/bluetooth/offBLECharacteristicValueChange.html>)
- 新增 API `ap.onBLEConnectionStateChanged()` 监听低功耗蓝牙连接的错误事件。详情 (<device/bluetooth/onBLEConnectionStateChanged.html>)
- 新增 API `ap.offBLEConnectionStateChanged()` 移除低功耗蓝牙连接的错误事件监听。详情 (<device/bluetooth/offBLEConnectionStateChanged.html>)
- 新增 API `ap.onBluetoothAdapterStateChange()` 监听本机蓝牙状态变化的事件。详情 (<device/bluetooth/onBluetoothAdapterStateChange.html>)
- 新增 API `ap.offBluetoothAdapterStateChange()` 移除本机蓝牙状态变化的事件监听。详情 (<device/bluetooth/offBluetoothAdapterStateChange.html>)

### **V0.3.1 更新日志** 2017-05-08

- 更新 API `ap.getLocation()`，默认入参 `type` 为 2，返回值新增速度和精度。详情 (<location/get/getLocation.html>)
- 修复 API `ap.getSystemInfo()`，调用出错的问题。

### **V0.3.0 更新日志** 2017-04-28

- 支持 `Promise` 语法。详情 (<index.html#快速开始>)
- 修复 API `ap.getStorage()` 获取字符串类型的数据时控制台报错的问题。

### **V0.2.9 更新日志** 2017-04-28

- • 修复 API `ap.getStorage()`，找不到对应 `key` 的 `data` 时，返回 `data` 为 `null`，并执行 `success` 回调。
- • 修复 API `ap.onPullDownRefresh()`，改为默认允许下拉刷新。

## **V0.2.8 更新日志** 2017-04-20

- 修复 API `ap.compressImage()`，压缩图片入参的问题。
- 修复 API `ap.getSystemInfo()`，获取窗口高度不正确的问题。

## **V0.2.7 更新日志** 2017-04-19

- 修复 API `ap.allowPullDownRefresh()` 禁用下拉刷新无效的问题。
- 修复 API `ap.makePhoneCall()` 拨打电话异常的问题。
- 更改 API `ap.setOptionButton()`，新增 `onClick` 入参，可以在 `OptionButton` 被点击时回调。详情 ([ui/navigationbar/setOptionButton.html](#))
- 新增 API `ap.compressImage()`，压缩图片。详情 ([media/image/compressImage.html](#))

## **V0.2.5 更新日志** 2017-03-31

- 修复 API `ap.setStorage()` 入参类型转换字符串的问题。
- 新增 API `ap.offSocketOpen()` 移除 `WebSocket` 链接打开事件监听。详情 ([network/webSocket/offSocketOpen.html](#))
- 新增 API `ap.offSocketClose()` 移除 `WebSocket` 关闭事件监听。详情 ([network/webSocket/offSocketClose.html](#))
- 新增 API `ap.offSocketError()` 移除 `WebSocket` 出错事件监听。详情 ([network/webSocket/offSocketError.html](#))
- 新增 API `ap.offSocketMessage()` 移除 `WebSocket` 消息事件监听。详情 ([network/webSocket/offSocketMessage.html](#))

## **V0.2.4 更新日志** 2017-03-17

- 更新 API `ap.getStorage()` 返回值。详情 ([data/storage/getStorage.html](#))
- 新增 API `ap.connectSocket()` 创建 `WebSocket` 链接。详情 ([network/webSocket/connectSocket.html](#))
- 新增 API `ap.sendSocketMessage()` 通过 `WebSocket` 连接发送数据。详情 ([network/webSocket/sendSocketMessage.html](#))
- 新增 API `ap.closeSocket()` 关闭 `WebSocket` 链接。详情 ([network/webSocket/closeSocket.html](#))
- 新增 API `ap.onSocketOpen()` `WebSocket` 链接打开事件。详情 ([network/webSocket/onSocketOpen.html](#))
- 新增 API `ap.onSocketMessage()` 接收到 `WebSocket` 消息事件。详情 ([network/webSocket/onSocketMessage.html](#))
- 新增 API `ap.onSocketError()` `WebSocket` 出错事件。详情 ([network/webSocket/onSocketError.html](#))
- 新增 API `ap.onSocketClose()` `WebSocket` 关闭事件。详情 ([network/webSocket/onSocketClose.html](#))

## **V0.2.3 更新日志** 2017-03-10

- • 新增 API `ap.onAppPause()` 监听应用压后台事件。详情 (navigation/appevent/onAppPause.html)
- • 新增 API `ap.offAppPause()` 移除应用压后台事件监听。详情 (navigation/appevent/offAppPause.html)
- • 新增 API `ap.onAppResume()` 监听应用唤起事件。详情 (navigation/appevent/onAppResume.html)
- • 新增 API `ap.offAppResume()` 移除应用唤起事件监听。详情 (navigation/appevent/offAppResume.html)
- • 新增 API `ap.setStorage()` 存储数据。详情 (data/storage/setStorage.html)
- • 新增 API `ap.getStorage()` 获取存储数据。详情 (data/storage/getStorage.html)
- • 新增 API `ap.removeStorage()` 删除存储数据。详情 (data/storage/removeStorage.html)
- • 更新 API `ap.share()` 参数列表。详情 (ui/share/share.html)

**V0.2.2 更新日志** 2017-03-06

- • 修复 API `ap.openLocation()` ，调用失败的问题。
- • 新增 API `ap.offBack()` 移除 back 事件监听。详情 (navigation/event/offBack.html)
- • 新增 API `ap.offPause()` 移除 pause 事件监听。详情 (navigation/event/offPause.html)
- • 新增 API `ap.offResume()` 移除 resume 事件监听。详情 (navigation/event/offResume.html)
- • 新增 API `ap.offTitleClick()` 移除标题点击事件监听。详情 (ui/navigationbar/offTitleClick.html)
- • 新增 API `ap.offPullDownRefresh()` 移除下拉刷新事件监听。详情 (ui/pulldown/offPullDownRefresh.html)
- • 新增 API `ap.offNetworkChange()` 移除网络状态变化事件监听。详情 (device/network/offNetworkChange.html)
- • 新增 API `ap.offBackgroundAudioPlay()` 移除音乐播放事件监听。详情 (media/audio/offBackgroundAudioPlay.html)
- • 新增 API `ap.offBackgroundAudioPause()` 移除音乐暂停事件监听。详情 (media/audio/offBackgroundAudioPause.html)
- • 新增 API `ap.offBackgroundAudioStop()` 移除音乐停止事件监听。详情 (media/audio/offBackgroundAudioStop.html)

**V0.2.1 更新日志** 2017-02-21

- • 新增 API `ap.onBackgroundAudioPlay()` 音乐播放事件。详情 (media/audio/onBackgroundAudioPlay.html)
- • 新增 API `ap.onBackgroundAudioPause()` 音乐暂停事件。详情 (media/audio/onBackgroundAudioPause.html)
- • 新增 API `ap.onBackgroundAudioStop()` 音乐停止事件。详情 (media/audio/onBackgroundAudioPause.html)

**V0.2.0 更新日志** 2017-02-16

- • 修复 页面导航类 API 在跳转时携带的 data 中有 Boolean 类型值丢失的问题。
- • 更新 API `ap.share()` ，增加 `showToolBar` 参数。详情 (ui/share/share.html)
- • 新增 API `ap.hideToast()` ，隐藏 toast。详情 (ui/notice/hideToast.html)

- • 新增 API `ap.openInBrowser()`，在浏览器打开网页。详情 (navigation/window/openInBrowser.html)
- • 新增 API `ap.openLocation()`，使用支付宝内置地图查看位置。详情 (location/open/openLocation.html)
- • 新增 API `ap.showPopMenu()`，在右上角显示弹出的下拉菜单。详情 (ui/navigationbar/showPopMenu.html)
- • 新增 API `ap.setOptionButton()`，设置导航栏右侧按钮。详情 (ui/navigationbar/setOptionButton.html)
- • 新增 API `ap.showOptionButton()`，显示导航栏右侧按钮。详情 (ui/navigationbar/showOptionButton.html)
- • 新增 API `ap.hideOptionButton()`，隐藏导航栏右侧按钮。详情 (ui/navigationbar/hideOptionButton.html)
- • 新增 API `ap.showBackButton()`，显示导航栏左侧返回按钮。详情 (ui/navigationbar/showBackButton.html)
- • 新增 API `ap.hideBackButton()`，隐藏导航栏左侧返回按钮。详情 (ui/navigationbar/hideBackButton.html)
- • 新增 API `ap.onBack()`，监听导航栏左侧返回按钮。详情 (navigation/event/onBack.html)
- • 新增 API `ap.allowBack()`，设置返回按钮及返回手势开关。详情 (ui/navigationbar/allowBack.html)
- • 整合 API `ap.enablePullDownRefresh()` 和 `ap.disablePullDownRefresh()` 为 `ap.allowPullDownRefresh()`。详情 (ui/pulldown/allowPullDownRefresh.html)
- • 新增 API `ap.makePhoneCall()`，拨打电话。详情 (device/call/makePhoneCall.html)
- • 新增 API `ap.startRecord()`，开始录音。详情 (media/record/startRecord.html)
- • 新增 API `ap.cancelRecord()`，取消录音。详情 (media/record/cancelRecord.html)
- • 新增 API `ap.stopRecord()`，停止录音。详情 (media/record/stopRecord.html)
- • 新增 API `ap.playVoice()`，开始播放语音。详情 (media/voice/playVoice.html)
- • 新增 API `ap.pauseVoice()`，暂停播放语音。详情 (media/voice/pauseVoice.html)
- • 新增 API `ap.stopVoice()`，停止播放语音。详情 (media/voice/stopVoice.html)
- • 新增 API `ap.playBackgroundAudio()` 音乐停止事件。详情 (media/audio/playBackgroundAudio.html)
- • 新增 API `ap.pauseBackgroundAudio()` 音乐停止事件。详情 (media/audio/pauseBackgroundAudio.html)
- • 新增 API `ap.seekBackgroundAudio()` 音乐停止事件。详情 (media/audio/seekBackgroundAudio.html)
- • 新增 API `ap.stopBackgroundAudio()` 音乐停止事件。详情 (media/audio/stopBackgroundAudio.html)
- • 新增 API `ap.getBackgroundAudioPlayerState()` 音乐状态信息。详情 (media/audio/getBackgroundAudioPlayerState.html)

**V0.1.2 更新日志** 2017-02-03

- • 更新 API `ap.onPullDownRefresh()` 新增 `refreshAvailable` 返回参数。详情 (event/onPullDownRefresh.html)
- • 修复 API `ap.showTabBar` 在 Android 上不触发回调的问题。
- • 修复 API `ap.datePicker` 在 Android 上返回的日期最后多个空格的问题。

# ap.parseQueryString( queryString )

把 url 中的 queryString 解析成一个对象。

## 1. 参数说明

| 参数          | 类型     | 必填 | 描述                              |
|-------------|--------|----|---------------------------------|
| queryString | String | 否  | 可不填，默认解析 window.location.search |

## 2. 返回值说明

| 类型     | 默认值 | 描述  |
|--------|-----|---|
| Object | {}  | queryString 解析结果。可配合 ap.pushWindow ( ../../navigation/window/pushWindow.html ) 等的 data 参数一起使用 |

## 3. 代码演示

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style> .output{ display:block; max-width: 100%; overflow: auto} </style>
<button id="J_btn" class="btn btn-default">parseQueryString</button>
<pre id="J_output" class="output"></pre>

<script>
  var btn = document.querySelector('#J_btn');
  var output = document.querySelector('#J_output');
  btn.addEventListener('click', function(){
    var params = ap.parseQueryString();
    output.innerHTML = '页面 URL:\n' +
                      window.location.href +
                      '\n解析结果:\n' +
                      JSON.stringify( params, undefined , '  ' );

  });

</script>
```

# ap.getSessionData(OPTION | keys | key, CALLBACK)

读取会话级数据。可用于页面间传递数据。

可直接传入一个数组作为 OPTION.keys，或直接传入一个字符串，作为 OPTION.keys 数组中的某一个 key。

## 1. OPTION 参数说明

| 名称   | 类型           | 必填 | 描述          |
|------|--------------|----|-------------|
| keys | String Array | 是  | 要读取的数据的 key |

## 2. CALLBACK 参数说明

| 名称   | 类型     | 描述             |
|------|--------|----------------|
| data | Object | keys 相应字段对应的数据 |

## 3. 代码演示

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn_get" style="display:none" class="btn btn-default">读取数据</button>
```

```
<button id="J_btn_set" style="display:none" class="btn btn-default">写入数据并打开新页面</button>
```

```
<script>
```

```
var btnSet= document.querySelector('#J_btn_set');
```

```
var btnGet = document.querySelector('#J_btn_get');
```

```
if (ap.parseQueryString().for === 'sessionData') {
```

```
    btnGet.style.display = 'block'
```

```
} else {
```

```
    btnSet.style.display = 'block';
```

```
}
```

```
btnSet.addEventListener('click', function(){
```

```
    ap.setSessionData({
```

```
        data: {
```

```
            cities: [
```

```
                {
```

```
                    cityName: '杭州',
```

```
                    adCode: '330100',
```

```
                    spell: ' hangzhou'
```

```
                }
```

```
            ],
```

```
            provinceName: '浙江省',
```

```
            countryName: '中国'
```

```
        }
```

```
    });
```

```
    setTimeout(function(){
```

```
        ap.pushWindow({
```

```
            url: 'getSessionData-demo.html',
```

```
            data: {
```

```
                for: 'sessionData'
```

```
            }
```

```
        });
```

```
    },100);
```

```
});
```

```
btnGet.addEventListener('click', function(){
```

```
    ap.getSessionData('cities', function(res) {
```

```
        ap.alert(res.data.cities[0].cityName);
```

```
    })
```

```
});
```

```
</script>
```

## ap.setSessionData(OPTION)



写入会话级数据。可直接传入一个对象作为 OPTION.data 参数。

## 1. OPTION 参数说明

| 名称   | 类型     | 必填 | 描述                            |
|------|--------|----|-------------------------------|
| data | Object | 是  | 要写入的数据，每个 key/value 的组合为一个数据项 |

## 2. 代码演示

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_set" class="btn btn-default">写入数据并打开新页面</button>
<script>
  var btnSet= document.querySelector('#J_btn_set');
  btnSet.addEventListener('click', function(){
    ap.setSessionData({
      data: {
        cities: [
          {
            cityName: '杭州',
            adCode: '330100',
            spell: ' hangzhou'
          }
        ],
        provinceName: '浙江省',
        countryName: '中国'
      }
    });
    setTimeout(function(){
      ap.pushWindow({
        url: 'getSessionData-demo.html',
        data: {
          for: 'sessionData'
        }
      });
    },100);
  });
</script>
```

# ap.offAccelerometerChange(CALLBACK)

停止监听重力感应变化

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
</style>
<h2 class="am-ft-center">晃晃手机试试</h2>
<pre id="J_output" class="output"></pre>
<script>
  var output = document.querySelector('#J_output');
  var handler = function(res){
    output.innerHTML = '\nx: ' + res.x +
                      '\ny: ' + res.y +
                      '\nz: ' + res.z;
    ap.offAccelerometerChange(handler);
  };
  ap.onAccelerometerChange(handler);
</script>
```

# ap.onAccelerometerChange(CALLBACK)

监听重力感应变化

## 1. CALLBACK 参数说明

| 名称 | 类型     | 描述  |
|----|--------|-----|
| x  | Number | X 轴 |
| y  | Number | Y 轴 |
| z  | Number | Z 轴 |

2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
</style>
<h2 class="am-ft-center">晃晃手机试试</h2>
<pre id="J_output" class="output"></pre>
<script>
  var output = document.querySelector('#J_output');
  ap.onAccelerometerChange(function(res){
    output.innerHTML = '\nx: ' + res.x +
                      '\ny: ' + res.y +
                      '\nz: ' + res.z;

  });
</script>
```

ap.closeBluetoothAdapter(OPTION, CALLBACK)

关闭本机蓝牙模块

1. 错误码说明

| error | 描述   |
|-------|------|
| 12    | 关闭失败 |

2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_open" class="btn btn-default">初始化本机蓝牙</button>
<button id="J_btn_close" class="btn btn-default">关闭本机蓝牙</button>
<script>
  var btnOpen = document.querySelector('#J_btn_open');
  var btnClose = document.querySelector('#J_btn_close');

  btnOpen.addEventListener('click', function(){
    ap.openBluetoothAdapter({
      success: function(res){
        if(!res.isSupportBLE) {
          ap.alert('抱歉，您的手机不支持 BLE');
        }
      },
      fail: function(res) {
        ap.showToast(res.errorMessage);
      }
    });
  });
  btnClose.addEventListener('click', function(){
    ap.closeBluetoothAdapter({
      fail: function(res){
        ap.showToast(res.errorMessage);
      }
    })
  });
});
</script>
```

# ap.connectBLEDevice(OPTION | deviceId, CALLBACK)

连接低功耗蓝牙设备。可直接传入一个字符串作为 OPTION.deviceId。

## 1. OPTION 参数说明

| 名称       | 类型     | 必填 | 描述      |
|----------|--------|----|---------|
| deviceId | String | 是  | 蓝牙设备 id |

## 2. 错误码说明

| error | 描述   |
|-------|------|
| 12    | 链接失败 |

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">连接BLE设备</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    //此处 deviceId 只是示例, 并非真正的设备 id
    ap.connectBLEDevice({
      deviceId: '123456789',
      success: function(res){
        ap.alert('链接成功');
      },
      fail: function(res){
        ap.showToast('链接失败-' + res.errorMessage);
      }
    });
  });
</script>
```

# ap.disconnectBLEDevice(OPTION | deviceId, CALLBACK)

断开与低功耗蓝牙设备的连接。可直接传入一个字符串作为 OPTION.deviceId。

### 1. OPTION 参数说明

| 名称       | 类型     | 必填 | 描述      |
|----------|--------|----|---------|
| deviceId | String | 是  | 蓝牙设备 id |

### 2. 错误码说明

|  |  |
|--|--|
|  |  |
|--|--|

| error | 描述   |
|-------|------|
| 12    | 断开失败 |

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">断开BLE设备</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    //此处 deviceId 只是示例, 并非真正的设备 id
    ap.disconnectBLEDevice({
      deviceId: '123456789',
      success: function(res){
        ap.alert('成功断开');
      },
      fail: function(res){
        ap.showToast('断开失败-' + res.errorMessage);
      }
    });
  });
</script>
```

## ap.getBLEDeviceCharacteristics(OPTION | deviceId, CALLBACK)

获取蓝牙设备所有 characteristic（特征值）。

### 1. OPTION 参数说明

| 名称        | 类型     | 必填 | 描述                    |
|-----------|--------|----|-----------------------|
| deviceId  | String | 是  | 蓝牙设备 id, 参考 device 对象 |
| serviceId | String | 是  | 蓝牙服务 uuid             |

### 2. CALLBACK 参数说明

| 名称              | 类型    | 描述      |
|-----------------|-------|---------|
| characteristics | Array | 设备特征值列表 |

## 2.1. characteristic 对象

蓝牙设备 characteristic （特征值）信息

| 名称               | 类型         | 描述                |
|------------------|------------|-------------------|
| characteristicId | String     | 蓝牙设备特征值的 uuid     |
| serviceId        | String     | 蓝牙设备特征值对应服务的 uuid |
| value            | Hex String | 蓝牙设备特征值对应的16进制值   |
| properties       | Object     | 该特征值支持的操作类型       |

## 2.2. properties 对象

| 名称       | 类型      | 描述                   |
|----------|---------|----------------------|
| 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 |

## 3. 错误码说明

| error | 描述   |
|-------|------|
| 12    | 获取失败 |

## 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">获取设备特征值</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.getBLEDeviceCharacteristics({
      // 这里的 deviceId 需要在 getBluetoothDevices 或 onBluetoothDeviceFound 接口中获取
      deviceId: deviceId,
      // 这里的 serviceId 需要在 getBLEDeviceServices 接口中获取
      serviceId: serviceId,
      success: function(res){
        ap.alert('此服务共有' + res.characteristics.length + '个特征值');
      },
      fail: function(res) {
        ap.toast('获取失败');
      }
    });
  });
</script>
```

# ap.getBLEDeviceServices(OPTION | deviceId, CALLBACK)

获取蓝牙设备所有 service（服务）。

## 1. OPTION 参数说明

| 名称       | 类型     | 必填 | 描述                    |
|----------|--------|----|-----------------------|
| deviceId | String | 是  | 蓝牙设备 id, 参考 device 对象 |

## 2. CALLBACK 参数说明

| 名称       | 类型    | 描述              |
|----------|-------|-----------------|
| services | Array | service 对象，详见下表 |

### 2.1. service对象



蓝牙设备service(服务)信息

| 名称        | 类型      | 描述           |
|-----------|---------|--------------|
| serviceId | String  | 蓝牙设备服务的 uuid |
| isPrimary | Boolean | 该服务是否为主服务    |

3. 错误码说明

| error | 描述   |
|-------|------|
| 12    | 获取失败 |

4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">获取设备服务</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.getBLEDeviceServices({
      // 这里的 deviceId 需要在 getBluetoothDevices 或 onBluetoothDeviceFound 接口中获取
      deviceId: deviceId,
      success: function(res){
        ap.alert('此设备共有' + res.services.length + '个服务');
      },
      fail: function(res) {
        ap.toast('获取失败');
      }
    });
  });
</script>
```

ap.getBluetoothAdapterState(OPTION, CALLBACK)

获取本机蓝牙模块状态

## 1. CALLBACK 参数说明

| 名称          | 类型      | 描述                          |
|-------------|---------|-----------------------------|
| discovering | Boolean | 是否正在搜索设备                    |
| available   | Boolean | 蓝牙模块是否可用(需支持 BLE 并且蓝牙是打开状态) |

## 2. 错误码说明

| error | 描述   |
|-------|------|
| 12    | 获取失败 |

## 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">获取本机蓝牙状态</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.getBluetoothAdapterState({
      success: function(res){
        if(!res.available) {
          ap.alert('抱歉，您的手机蓝牙暂不可用');
        }
        if(res.discovering) {
          ap.showToast('正在搜索蓝牙设备');
        }
      },
      fail: function(res) {
        ap.showToast(res.errorMessage);
      }
    });
  });
}</script>
```

ap.getBluetoothDevices(OPTION | services | service, CALLBACK)

获取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。可直接传入一个数组作为 `OPTION.services` 参数。也可直接传入一个字符串作为 `OPTION.services` 的第一项。

## 1. OPTION 参数说明

| 名称       | 类型    | 必填 | 描述                      |
|----------|-------|----|-------------------------|
| services | Array | 否  | 蓝牙设备主 service 的 uuid 列表 |

## 2. CALLBACK 参数说明

| 名称      | 类型    | 描述                                     |
|---------|-------|--|
| devices | Array | service 的 uuid 对应的(如果有传入 services)设备列表 |

### 2.1. device对象说明

| 名称           | 类型         | 描述              |
|--------------|------------|-----------------|
| deviceName   | String     | 蓝牙设备名称，某些设备可能没有 |
| localName    | String     | 广播设备名称          |
| deviceId     | String     | 设备 Id           |
| RSSI         | Number     | 设备信号强度          |
| advertisData | Hex String | 设备的广播内容         |

## 3. 错误码说明

| error | 描述   |
|-------|------|
| 12    | 获取失败 |

## 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn" class="btn btn-default">获取已发现的蓝牙设备</button>
```

```
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.getBluetoothDevices({
      success: function(res){
        ap.alert('共发现' + res.devices.length + '个设备');
      }
    });
  });
}</script>
```

# ap.getConnectedBluetoothDevices(OPTION | services | service, CALLBACK)

根据 service 的 uuid 获取处于已连接状态的设备。可直接传入一个数组作为 OPTION.services 参数。也可直接传入一个字符串作为 OPTION.services 的第一项。

## 1. OPTION 参数说明

| 名称       | 类型    | 必填 | 描述                      |
|----------|-------|----|-------------------------|
| services | Array | 否  | 蓝牙设备主 service 的 uuid 列表 |

## 2. CALLBACK 参数说明

| 名称      | 类型    | 描述                                     |
|---------|-------|--|
| devices | Array | service 的 uuid 对应的(如果有传入 services)设备列表 |

### 2.1. device对象说明

| 名称         | 类型     | 描述              |
|------------|--------|-----------------|
| deviceName | String | 蓝牙设备名称，某些设备可能没有 |
| localName  | String | 广播设备名称          |
|            |        |                 |

|              |            |         |
|--------------|------------|---------|
| deviceId     | String     | 设备 Id   |
| RSSI         | Number     | 设备信号强度  |
| advertisData | Hex String | 设备的广播内容 |

### 3. 错误码说明

| error | 描述   |
|-------|------|
| 12    | 获取失败 |

### 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">获取已连接的蓝牙设备</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.getConnectedBluetoothDevices({
      success: function(res){
        ap.alert('共发现' + res.devices.length + '个设备');
      }
    });
  });
</script>
```

## ap.notifyBLECharacteristicValueChange(OPTION, CALLBACK)

启用低功耗蓝牙设备特征值变化时的 notify 功能。注意：必须设备的特征值支持notify才可以成功调用，具体参照 characteristic 的 properties 属性 另外，必须先启用 notify才能监听到设备 characteristicValueChange 事件。

### 1. OPTION 参数说明

| 名称 | 类型 | 必填 | 描述 |
|----|----|----|----|
|    |    |    |    |

|                  |        |   |                            |
|------------------|--------|---|----------------------------|
| deviceId         | String | 是 | 蓝牙设备 id, 参考 device 对象      |
| serviceId        | String | 是 | 蓝牙特征值对应 service 的 uuid     |
| characteristicId | String | 是 | 蓝牙特征值的 uuid                |
| descriptorId     | String | 否 | notify 的 descriptor 的 uuid |

## 2. 错误码说明

| error | 描述   |
|-------|------|
| 12    | 启动失败 |

## 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">启用 notify 功能</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.notifyBLECharacteristicValueChange({
      // 这里的 deviceId 需要在 getBluetoothDevices 或 onBluetoothDeviceFound 接口中获取
      deviceId: deviceId,
      // 这里的 serviceId 需要在 getBLEDeviceServices 接口中获取
      serviceId: serviceId,
      // 这里的 characteristicId 需要在 getBLEDeviceCharacteristics 接口中获取
      characteristicId: characteristicId,
      success: function(res){
        ap.alert('启动通知成功');
      },
      fail: function(res) {
        ap.toast('启动通知失败');
      }
    });
  });
</script>
```

# ap.offBLECharacteristicValueChange(CALLBACK)

移除低功耗蓝牙设备的特征值变化事件的监听。

# ap.offBLEConnectionStateChanged(CALLBACK)

移除低功耗蓝牙连接的错误事件的监听。

# ap.offBluetoothAdapterStateChange(CALLBACK)

移除本机蓝牙状态变化的事件的监听。

# ap.offBluetoothDeviceFound(CALLBACK)

移除寻找到新的蓝牙设备事件的监听。

# ap.onBLECharacteristicValueChange(CALLBACK)

监听低功耗蓝牙设备的特征值变化的事件。

## 1. CALLBACK 参数说明

| 名称 | 类型 | 描述 |
|----|----|----|
|    |    |    |

|                  |            |                       |
|------------------|------------|-----------------------|
| deviceId         | String     | 蓝牙设备 id, 参考 device 对象 |
| serviceId        | String     | 特征值所属服务 uuid          |
| characteristicId | String     | 特征值 uuid              |
| value            | Hex String | 特征值最新的16进制值           |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">启用 notify 功能</button>
<script>
  var btn = document.querySelector('#J_btn');
  ap.onBLECharacteristicValueChange(function(res){
    var msg = '特征值变化: ' + value;
    ap.showToast(msg);
  });
  btn.addEventListener('click', function(){
    ap.notifyBLECharacteristicValueChange({
      // 这里的 deviceId 需要在 getBluetoothDevices 或 onBluetoothDeviceFound 接口中获取
      deviceId: deviceId,
      // 这里的 serviceId 需要在 getBLEDeviceServices 接口中获取
      serviceId: serviceId,
      // 这里的 characteristicId 需要在 getBLEDeviceCharacteristics 接口中获取
      characteristicId: characteristicId,
      success: function(res){
        ap.alert('启动通知成功');
      },
      fail: function(res) {
        ap.toast('启动通知失败');
      }
    });
  });
</script>
```

# ap.onBLEConnectionStateChanged(CALLBACK)

监听低功耗蓝牙连接的错误事件，包括设备丢失，连接异常断开等等。



## 1. CALLBACK 参数说明

| 名称        | 类型      | 描述                    |
|-----------|---------|-----------------------|
| deviceId  | String  | 蓝牙设备 id, 参考 device 对象 |
| connected | Boolean | 连接目前的状态               |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">监听蓝牙设备事件</button>
<script>
  var btn = document.querySelector('#J_btn');

  btn.addEventListener('click', function(){
    ap.onBLEConnectionStateChanged(function(res){
      var msg = '设备 id: ' + res.deviceId +
                '是否链接: ' + res.connected;
      ap.showToast(msg);
    });
  });
</script>
```

# ap.onBluetoothAdapterStateChange(CALLBACK)

监听本机蓝牙状态变化的事件。

## 1. CALLBACK 参数说明

| 名称          | 类型      | 描述           |
|-------------|---------|--------------|
| available   | Boolean | 蓝牙模块是否可用     |
| discovering | Boolean | 蓝牙模块是否处于搜索状态 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">监听本机蓝牙状态</button>
<script>
  var btn = document.querySelector('#J_btn');

  btn.addEventListener('click', function(){
    ap.onBluetoothAdapterStateChange(function(res){
      var msg = '蓝牙是否可用: ' + res.available +
        '是否在搜索: ' + discovering;
      ap.alert(msg);
    });
  });
</script>
```

# ap.onBluetoothDeviceFound(CALLBACK)

寻找到新的蓝牙设备时触发此事件。

## 1. CALLBACK 参数说明

| 名称      | 类型    | 描述                       |
|---------|-------|--------------------------|
| devices | Array | 新搜索到的设备列表。device 对象说明见下表 |

### device 对象说明

| 名称           | 类型         | 描述              |
|--------------|------------|-----------------|
| deviceName   | String     | 蓝牙设备名称，某些设备可能没有 |
| localName    | String     | 广播设备名称          |
| deviceId     | String     | 设备 Id           |
| RSSI         | Number     | 设备信号强度          |
| advertisData | Hex String | 设备的广播内容         |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">开始搜索周围蓝牙设备</button>
<script>
  var btn = document.querySelector('#J_btn');
  ap.onBluetoothDeviceFound(function(res){
    ap.showToast('搜索到' + res.devices.length + '个设备');
  });
  btn.addEventListener('click', function(){
    //此处 service 只是示例, 并非真正的设备服务 id
    ap.startBluetoothDevicesDiscovery('0000fdd7-0000-1011-8004-00987f9b34fb');
  });
</script>
```

# ap.openBluetoothAdapter(OPTION, CALLBACK)

初始化本机蓝牙模块。

## 1. CALLBACK 参数说明

| 名称           | 类型      | 描述       |
|--------------|---------|----------|
| isSupportBLE | Boolean | 是否支持 BLE |

## 2. 错误码说明

| error | 描述           |
|-------|--------------|
| 12    | 蓝牙未打开        |
| 13    | 与系统服务的链接暂时丢失 |
| 14    | 未授权支付宝使用蓝牙功能 |
| 15    | 未知错误         |

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">初始化本机蓝牙</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.openBluetoothAdapter({
      success: function(res){
        if(!res.isSupportBLE) {
          ap.alert('抱歉，您的手机不支持 BLE');
        }
      },
      fail: function(res) {
        ap.showToast(res.errorMessage);
      }
    });
  });
</script>
```

## ap.readBLECharacteristicValue(OPTION, CALLBACK)

读取低功耗蓝牙设备特征值中的数据。调用后在 ap.onBLECharacteristicValueChange() 事件中接收数据返回。

### 1. OPTION 参数说明

| 名称               | 类型     | 必填 | 描述                         |
|------------------|--------|----|----------------------------|
| deviceId         | String | 是  | 蓝牙设备 id, 参考 device 对象      |
| serviceId        | String | 是  | 蓝牙特征值对应 service 的 uuid     |
| characteristicId | String | 是  | 蓝牙特征值的 uuid                |
| descriptorId     | String | 否  | notify 的 descriptor 的 uuid |

### 2. CALLBACK 参数说明

| 名称 | 类型 | 描述 |
|----|----|----|
|----|----|----|

|                |        |         |
|----------------|--------|---------|
| characteristic | Object | 设备特征值信息 |
|----------------|--------|---------|

## 2.1. characteristic对象

蓝牙设备characteristic(特征值)信息

| 名称               | 类型     | 描述                |
|------------------|--------|-------------------|
| characteristicId | String | 蓝牙设备特征值的 uuid     |
| serviceId        | String | 蓝牙设备特征值对应服务的 uuid |

## 3. 错误码说明

| error | 描述     |
|-------|--------|
| 12    | 读取数据失败 |

## 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">读取设备特征值</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.readBLECharacteristicValue({
      // 这里的 deviceId 需要在 getBluetoothDevices 或 onBluetoothDeviceFound 接口中获取
      deviceId: deviceId,
      // 这里的 serviceId 需要在 getBLEDeviceServices 接口中获取
      serviceId: serviceId,
      // 这里的 characteristicId 需要在 getBLEDeviceCharacteristics 接口中获取
      characteristicId: characteristicId,
      success: function(res){
        ap.alert('正在读取, 请在 ap.onBLECharacteristicValueChange 事件中接收');
      },
      fail: function(res) {
        ap.toast('读取数据指令失败');
      }
    });
  });
</script>
```

# ap.startBluetoothDevicesDiscovery(OPTION | services | service, CALLBACK)

开始搜寻附近的蓝牙外围设备。搜索结果将在 bluetoothDeviceFound 事件中返回。可直接传入一个数组作为 OPTION.services 参数。也可直接传入一个字符串作为 OPTION.services 的第一项。

## 1. OPTION 参数说明

| 名称       | 类型    | 必填 | 描述                      |
|----------|-------|----|-------------------------|
| services | Array | 否  | 蓝牙设备主 service 的 uuid 列表 |

## 2. CALLBACK 参数说明

| 名称       | 类型    | 描述     |
|----------|-------|--------|
| services | Array | 扫码所得数据 |

### 3. 错误码说明

| error | 描述     |
|-------|--------|
| 12    | 开始搜索失败 |

### 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">开始搜索周围蓝牙设备</button>
<script>
  var btn = document.querySelector('#J_btn');
  ap.onBluetoothDeviceFound(function(res){
    ap.showToast('搜索到' + res.devices.length + '个设备');
  });
  btn.addEventListener('click', function(){
    //此处 service 只是示例, 并非真正的设备服务 id
    ap.startBluetoothDevicesDiscovery({
      services: ['0000fdd7-0000-1011-8004-00987f9b34fb']
    });
  });
</script>
```

## ap.stopBluetoothDevicesDiscovery()

停止搜寻附近的蓝牙外围设备。

### 1. 错误码说明

| error | 描述     |
|-------|--------|
| 12    | 停止搜索失败 |

### 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_start" class="btn btn-default">开始搜索周围蓝牙设备</button>
<button id="J_btn_stop" class="btn btn-default">停止搜索周围蓝牙设备</button>
<script>
  var btnStart = document.querySelector('#J_btn_start');
  var btnStop = document.querySelector('#J_btn_stop');
  ap.onBluetoothDeviceFound(function(res){
    ap.showToast('搜索到' + res.devices.length + '个设备');
  });
  btnStart.addEventListener('click', function(){
    //此处 service 只是示例, 并非真正的设备服务 id
    ap.startBluetoothDevicesDiscovery('0000fdd7-0000-1011-8004-00987f9b34fb');
  });
  btnStop.addEventListener('click', function(){
    ap.stopBluetoothDevicesDiscovery();
  });
</script>
```

# ap.writeBLECharacteristicValue(OPTION, CALLBACK)

向低功耗蓝牙设备特征值中写入数据。

## 1. OPTION 参数说明

| 名称               | 类型         | 必填 | 描述                         |
|------------------|------------|----|----------------------------|
| deviceId         | String     | 是  | 蓝牙设备 id, 参考 device 对象      |
| serviceId        | String     | 是  | 蓝牙特征值对应 service 的 uuid     |
| characteristicId | String     | 是  | 蓝牙特征值的 uuid                |
| descriptorId     | String     | 否  | notify 的 descriptor 的 uuid |
| value            | Hex String | 是  | 蓝牙设备特征值对应的值, 16进制字符串       |

## 2. 错误码说明

| error | 描述 |
|-------|----|
|-------|----|



### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">向设备写入特征值</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.writeBLECharacteristicValue({
      // 这里的 deviceId 需要在上面的 getBluetoothDevices 或 onBluetoothDeviceFound 接口中获取
      deviceId: deviceId,
      // 这里的 serviceId 需要在 getBLEDeviceServices 接口中获取
      serviceId: serviceId,
      // 这里的 characteristicId 需要在 getBLEDeviceCharacteristics 接口中获取
      characteristicId: characteristicId,
      // 这里的value是 16 进制字符串
      value: value,
      success: function(res){
        ap.alert('写入数据成功');
      },
      fail: function(res) {
        ap.toast('写入数据失败');
      }
    });
  });
</script>
```

## ap.makePhoneCall(OPTION | number)

拨打电话。可直接传入一个字符串作为 OPTION.number 参数。

### 1. OPTION 参数说明

| 名称     | 类型     | 必填 | 描述   |
|--------|--------|----|------|
| number | String | 是  | 电话号码 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">拨打支付宝客服</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.makePhoneCall('95188');
  });
</script>
```

# ap.offCompassChange(CALLBACK)

停止监听罗盘数据变化

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
</style>
<h2 class="am-ft-center">转转手机试试</h2>
<pre id="J_output" class="output"></pre>
<script>
  var output = document.querySelector('#J_output');
  var handler = function(res){
    output.innerHTML = 'direction: ' + res.direction;
    ap.offCompassChange(handler);
  };
  ap.onCompassChange(handler);
</script>
```

# ap.onCompassChange(CALLBACK)

## 1. CALLBACK 参数说明

| 名称        | 类型     | 描述   |
|-----------|--------|------|
| direction | Number | 方向度数 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
</style>
<h2 class="am-ft-center">转转手机试试</h2>
<pre id="J_output" class="output"></pre>
<script>
  var output = document.querySelector('#J_output');
  ap.onCompassChange(function(res){
    output.innerHTML = 'direction: ' + res.direction
  });
</script>
```

# ap.getNetworkType(OPTION, CALLBACK)

获取当前网络状态

## 1. CALLBACK 参数说明

| 名称               | 类型      | 描述  |
|------------------|---------|---|
| networkAvailable | Boolean | 网络是否可用  |
| networkType      | String  | 网络类型值 UNKNOWN / NOTREACHABLE / WIFI / 3G / 2G / 4G / WWAN |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_network" class="btn btn-default">获取当前网络类型</button>
<script>
  var btnNetwork = document.querySelector('#J_btn_network');
  btnNetwork.addEventListener('click', function(){
    ap.getNetworkType(function(res){
      ap.alert({
        title: '网络' + (res.networkAvailable ? '畅通' : '不通'),
        content: '网络类型: ' + res.networkType
      });
    });
  });
</script>
```

# ap.offNetworkChange(CALLBACK)

停止监听网络环境发生变化

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2 class="am-ft-center">切换网络试试</h2>
<script>
  var handler = function(res){
    ap.alert('当前网络类型: ' + res.networkType);
    ap.offNetworkChange(handler);
  };
  ap.onNetworkChange(handler);
</script>
```

# ap.onNetworkChange(CALLBACK)

监听网络环境发生变化

## 1. CALLBACK 参数说明

和 `ap.getNetworkType()` 返回值一致

| 名称               | 类型      | 描述  |
|------------------|---------|---|
| networkAvailable | Boolean | 网络是否可用  |
| networkType      | String  | 网络类型值 UNKNOWN / NOTREACHABLE / WIFI / 3G / 2G / 4G / WWAN |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2 class="am-ft-center">切换网络试试</h2>
<script>
  ap.onNetworkChange(function(res){
    ap.alert('当前网络类型: ' + res.networkType);
  });
</script>
```

## 3. 其他说明

- 3G切换到4G时系统不会发出网络切换通知。 4G切到3G时，会先从4G跳到2G，然后再从2G跳3G，因此会通知两次

# ap.scan(OPTION | type, CALLBACK)

调用扫一扫功能

## 1. OPTION 参数说明

| 名称   | 类型     | 必填 | 描述                                 |
|------|--------|----|------------------------------------|
| type | String | 否  | 扫描目标类型，支持 qr / bar，相应扫码选框会不同，默认 qr |

## 2. CALLBACK 参数说明

| 名称   | 类型     | 描述     |
|------|--------|--------|
| code | String | 扫码所得数据 |

## 3. 错误码说明

| error | 描述   |
|-------|------|
| 10    | 用户取消 |
| 11    | 操作失败 |

## 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_scanQR" class="btn btn-default">扫一扫二维码</button>
<button id="J_btn_scanBAR" class="btn btn-default">扫一扫条形码</button>
<script>
  var btnScanQR = document.querySelector('#J_btn_scanQR');
  var btnScanBAR = document.querySelector('#J_btn_scanBAR');
  btnScanQR.addEventListener('click', function(){
    ap.scan(function(res){
      ap.alert(res.code);
    });
  });
  btnScanBAR.addEventListener('click', function(){
    ap.scan({
      type: 'bar'
    }, function(res){
      ap.alert(res.code);
    });
  });
</script>
```

# ap.watchShake(OPTION, CALLBACK)

摇一摇功能。每次调用API，在摇一摇手机后触发回调，如需再次监听需要再次调用这个API。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2 class="am-ft-center">摇一摇手机试试</h2>
<script>
  (function watchShake(){
    ap.watchShake(function(res){
      ap.confirm({
        content: '亲，还摇不? ',
        cancelButtonText: '不摇了',
        confirmButtonText: '继续摇'
      }, function(res){
        if (res.confirm) {
          // 再次调用才能继续监听摇一摇动作
          watchShake();
        }
      });
    });
  })();
</script>
```

# ap.vibrate()

调用震动功能

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn_vibrate" class="btn btn-default">点击感受震动</button>
```

```
<script>
  var btnVibrate = document.querySelector('#J_btn_vibrate');
  btnVibrate.addEventListener('click', function(){
    ap.vibrate();
  });
</script>
```

## 1. 关于 AlipayJSAPI

AlipayJSAPI 提供丰富的支付宝原生 API，可以方便调用支付宝提供的能力，达到媲美原生应用的体验，如页面跳转，支付功能等。

### 1.1. 安装使用

- 页面引入：

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.min.js"></script>
```

### 1.2. 快速开始

- **Promise 方式：**

```
ap.chooseImage()
  .then(function(result){
    return ap.uploadFile({
      url: '[请使用自己服务器地址]',
      fileType: 'image',
      fileName: 'file',
      filePath: result.apFilePaths[0]
    });
  })
  .then(function(result){
    return ap.alert('上传成功');
  })
  .catch(function(){
    ap.showToast('上传失败');
  });
```

- 回调方式：



```
ap.chooseImage(function(res){
  if(res.apFilePaths && res.apFilePaths.length > 0) {
    ap.uploadFile({
      url: '[请使用自己服务器地址]',
      fileType: 'image',
      fileName: 'file',
      filePath: res.apFilePaths[0],
      success: function() {
        ap.alert('上传成功');
      },
      fail: function() {
        ap.showToast('上传失败');
      }
    });
  }
});
```

```
ap.confirm('Hello AlipayJSAPI!', function(result){
  ap.showToast(result.confirm ? '确定' : '取消');
});
```

## 2. 其他说明

### 2.1. 接口约定

- ap.on 开头的 API 是监听某个事件发生的接口，接受一个 CALLBACK 函数作为参数。当该事件触发时，会调用 CALLBACK 函数。
- ap.off 开头的 API 是移除相应事件监听函数的接口，若传入事件注册（ap.on）时的 CALLBACK 函数，即移除指定的监听函数。若不传入则移除此事件所有监听函数。
- 如未特殊约定，其他 API 接口都接收两个参数，一个 Object 类型的 OPTION 作为入参，OPTION 中可以指定 success / fail / complete 来接收接口调用结果；还有一个可选参数是 Function 类型的 CALLBACK，作为回调函数等同于 OPTION.complete。接口参数在文档里没有明确定义时，则无须传入。
- 回调函数的参数始终是 Object 类型，会返回接口调用的结果。注意，并不是每个接口都有回调函数，可在具体接口文档里查看是否定义。
- 有回调函数的接口均支持 Promise 风格调用。

| 名称       | 类型       | 必填 | 描述                       |
|----------|----------|----|--------------------------|
| 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） |

## 2.2. 错误约定

AlipayJSAPI 调用出错时 CALLBACK 回调函数的参数里会包含一个特殊的字段 error（Number 类型）， 作为 API 调用的错误码， 示例:

```
{
  error: 1,
  errorMessage: '接口不存在'
}
```

| error | 描述     |
|-------|--------|
| 1     | 接口不存在  |
| 2     | 参数无效   |
| 3     | 发生未知错误 |
| 4     | 接口无权限  |

# ap.getLocation(CALLBACK)

获取用户当前的地理位置信息

## 1. OPTION 参数说明

| 名称           | 类型     | 必填 | 描述  |
|--------------|--------|----|---|
| cacheTimeout | Number | 否  | 钱包经纬度定位缓存过期时间，单位秒。默认 30s。使用缓存会加快定位速度，缓存过期会重新定位    |
| type         | Number | 否  | 支持 0：详细逆地理编码，带周边信；1：逆地理编码到城市；2：仅获取经纬度、速度和精度。默认为 2 |
| timeout      | Number | 否  | 定位超时失败回调时间，单位秒。默认10s                              |
| bizType      | String | 否  | 自定义业务类型   |

## 2. CALLBACK 参数说明

- 当 type 为 0 时，会返回下面全部字段。

| 名称           | 类型           | 描述                               |
|--------------|--------------|----------------------------------|
| longitude    | String       | 经度                               |
| latitude     | String       | 纬度                               |
| accuracy     | Number       | 精度，单位米                           |
| speed        | Number       | 速度，单位毫秒                          |
| country      | String       | 国家名                              |
| countryCode  | string       | 国家编号                             |
| province     | String       | 省份名                              |
| city         | String       | 城市名                              |
| cityCode     | String       | 城市编码                             |
| adCode       | String       | 区域编码                             |
| streetNumber | Object       | 街道门牌信息，结构是：{street，number}       |
| pois         | Object Array | 定位点附近的 POI 信息，结构是：{name，address} |

### 3. 错误码说明

| error | 描述           |
|-------|--------------|
| 12    | GPS打开，但定位失败  |
| 13    | 获取地理位置信息失    |
| 14    | 定位超时         |
| 15    | 网络错误         |
| 16    | GPS未打开或用户未授权 |

### 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
</style>
<button id="J_btn_location" class="btn btn-default">获取当前位置信息</button>
<pre id="J_output" class="output"></pre>
<script>
  var btnLocation = document.querySelector('#J_btn_location');
  var output = document.querySelector('#J_output');
  btnLocation.addEventListener('click', function(){
    ap.getLocation(function(res) {
      output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
  });
</script>
```

5. 其他说明

# ap.openLocation(OPTION, CALLBACK)

使用支付宝内置地图查看位置。

## 1. OPTION 参数说明

| 名称        | 类型     | 必填 | 描述                 |
|-----------|--------|----|--------------------|
| longitude | String | 是  | 经度                 |
| latitude  | String | 是  | 纬度                 |
| name      | String | 否  | 位置名称               |
| address   | String | 否  | 地址的详细说明            |
| scale     | Number | 否  | 缩放比例， 范围3~19，默认为15 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn" class="btn btn-default">openLocation</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.openLocation({
      longitude: '121.549697',
      latitude: '31.227250',
      name: '支付宝',
      address: '杨高路地铁站'
    });
  });
</script>
```

# ap.getBackgroundAudioPlayerState(CALLBACK)

使用后台播放器播放音乐。

## 1. CALLBACK 参数说明

| 名称              | 类型     | 描述                            |
|-----------------|--------|-------------------------------|
| status          | Number | 音乐播放状态（2：没有音乐在播放，1：播放中，0：暂停中） |
| duration        | Number | 音乐总时长，单位秒                     |
| currentPosition | Number | 当前播放位置，单位秒                    |
| downloadPercent | Number | 缓冲百分比                         |
| url             | String | 音乐链接地址                        |

## 2. 错误码描述

| error | 描述        |
|-------|-----------|
| 12    | 失败时会有对应描述 |

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<pre id="J_output" class="output"></pre>
<script>
  var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
  var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
  var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
  var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
  var btnGetState = document.querySelector('#J_btnGetState');
  var output = document.querySelector('#J_output');

  btnPlayBackgroundAudio.addEventListener('click', function(){
    ap.playBackgroundAudio({
      url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflldZgjBdxE.mp3',
      title: '舒缓轻音乐',
      cover: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png'
    });
  });
  btnPauseBackgroundAudio.addEventListener('click', function(){
    ap.pauseBackgroundAudio();
  });
  btnStopBackgroundAudio.addEventListener('click', function(){
    ap.stopBackgroundAudio();
  });
  btnSeekBackgroundAudio.addEventListener('click', function(){
    ap.seekBackgroundAudio(0);
  });
  btnGetState.addEventListener('click', function(){
    ap.getBackgroundAudioPlayerState(function(res){
      output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
  });
</script>
```

### 4. 其他说明

- 此接口1 0.0.3 以上才支持

# ap.offBackgroundAudioPause(CALLBACK)

停止监听音乐暂停事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<button id="J_btnOn" class="btn btn-default">开启事件监听</button>
<button id="J_btnOff" class="btn btn-default">关闭事件监听</button>

<pre id="J_output" class="output"></pre>
<script>
  var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
  var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
  var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
  var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
  var btnGetState = document.querySelector('#J_btnGetState');
  var btnOn = document.querySelector('#J_btnOn');
  var btnOff = document.querySelector('#J_btnOff');
  var output = document.querySelector('#J_output');

  var handlerPlay = function(){
    ap.showToast('开始播放')
  };
  var handlerPause = function(){
    ap.showToast('暂停播放')
  };
  var handlerStop = function(){
    ap.showToast('停止播放')
  };

  btnPlayBackgroundAudio.addEventListener('click', function(){
    ap.playBackgroundAudio({
      title: '舒缓轻音乐',
      url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflZgjBdxE.mp3',
      logo: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png',
      cover: 'https://zos.alipayobjects.com/rmsportal/SKMizgNnNTQNdJWzQfdI.JPEG'
```

```
});
});
btnPauseBackgroundAudio.addEventListener('click', function(){
    ap.pauseBackgroundAudio();
});
btnStopBackgroundAudio.addEventListener('click', function(){
    ap.stopBackgroundAudio();
});
btnSeekBackgroundAudio.addEventListener('click', function(){
    ap.seekBackgroundAudio(0);
});
btnGetState.addEventListener('click', function(){
    ap.getBackgroundAudioPlayerState(function(res){
        output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
});
btnOn.addEventListener('click', onEvents);
btnOff.addEventListener('click', offEvents);

function onEvents() {
    ap.onBackgroundAudioPlay(handlerPlay);
    ap.onBackgroundAudioPause(handlerPause);
    ap.onBackgroundAudioStop(handlerStop);
}
function offEvents() {
    ap.offBackgroundAudioPlay(handlerPlay);
    ap.offBackgroundAudioPause(handlerPause);
    ap.offBackgroundAudioStop(handlerStop);
}
</script>
```

## 2. 其他说明

- 此接口10.0.3 以上才支持

# ap.offBackgroundAudioPlay(CALLBACK)

停止监听音乐播放事件。

## 1. 代码示例



```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<button id="J_btnOn" class="btn btn-default">开启事件监听</button>
<button id="J_btnOff" class="btn btn-default">关闭事件监听</button>
```

```
<pre id="J_output" class="output"></pre>
```

```
<script>
```

```
var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
var btnGetState = document.querySelector('#J_btnGetState');
var btnOn = document.querySelector('#J_btnOn');
var btnOff = document.querySelector('#J_btnOff');
var output = document.querySelector('#J_output');
```

```
var handlerPlay = function(){
    ap.showToast('开始播放')
};
var handlerPause = function(){
    ap.showToast('暂停播放')
};
var handlerStop = function(){
    ap.showToast('停止播放')
};
```

```
btnPlayBackgroundAudio.addEventListener('click', function(){
    ap.playBackgroundAudio({
        title: '舒缓轻音乐',
        url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflZgjBdxE.mp3',
        logo: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png',
        cover: 'https://zos.alipayobjects.com/rmsportal/SKMIZgNnNTQNdJWzQfdI.JPEG'
    });
});
btnPauseBackgroundAudio.addEventListener('click', function(){
    ap.pauseBackgroundAudio();
});
btnStopBackgroundAudio.addEventListener('click', function(){
    ap.stopBackgroundAudio();
});
btnSeekBackgroundAudio.addEventListener('click', function(){
    ap.seekBackgroundAudio(0);
});
```

```
btnGetState.addEventListener('click', function(){
    ap.getBackgroundAudioPlayerState(function(res){
        output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
});
btnOn.addEventListener('click', onEvents);
btnOff.addEventListener('click', offEvents);

function onEvents() {
    ap.onBackgroundAudioPlay(handlerPlay);
    ap.onBackgroundAudioPause(handlerPause);
    ap.onBackgroundAudioStop(handlerStop);
}
function offEvents() {
    ap.offBackgroundAudioPlay(handlerPlay);
    ap.offBackgroundAudioPause(handlerPause);
    ap.offBackgroundAudioStop(handlerStop);
}
</script>
```

## 2. 其他说明

- 此接口10.0.3 以上才支持

# ap.offBackgroundAudioStop(CALLBACK)

停止监听音乐停止事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<button id="J_btnOn" class="btn btn-default">开启事件监听</button>
<button id="J_btnOff" class="btn btn-default">关闭事件监听</button>
```

```
<pre id="J_output" class="output"></pre>
<script>
    var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
    var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
    var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
    var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
    var btnGetState = document.querySelector('#J_btnGetState');
    var btnOn = document.querySelector('#J_btnOn');
    var btnOff = document.querySelector('#J_btnOff');
    var output = document.querySelector('#J_output');

    var handlerPlay = function(){
        ap.showToast('开始播放')
    };
    var handlerPause = function(){
        ap.showToast('暂停播放')
    };
    var handlerStop = function(){
        ap.showToast('停止播放')
    };

    btnPlayBackgroundAudio.addEventListener('click', function(){
        ap.playBackgroundAudio({
            title: '舒缓轻音乐',
            url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflDZgjBdxE.mp3',
            logo: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png',
            cover: 'https://zos.alipayobjects.com/rmsportal/SKMizgNnNTQNdJWzQfdI.JPEG'
        });
    });
    btnPauseBackgroundAudio.addEventListener('click', function(){
        ap.pauseBackgroundAudio();
    });
    btnStopBackgroundAudio.addEventListener('click', function(){
        ap.stopBackgroundAudio();
    });
    btnSeekBackgroundAudio.addEventListener('click', function(){
        ap.seekBackgroundAudio(0);
    });
    btnGetState.addEventListener('click', function(){
        ap.getBackgroundAudioPlayerState(function(res){
            output.innerHTML = JSON.stringify(res, undefined, '  ');
        });
    });
    btnOn.addEventListener('click', onEvents);
    btnOff.addEventListener('click', offEvents);

    function onEvents() {
        ap.onBackgroundAudioPlay(handlerPlay);
        ap.onBackgroundAudioPause(handlerPause);
        ap.onBackgroundAudioStop(handlerStop);
    }
</script>
```

```
}
function offEvents() {
    ap.offBackgroundAudioPlay(handlerPlay);
    ap.offBackgroundAudioPause(handlerPause);
    ap.offBackgroundAudioStop(handlerStop);
}
</script>
```

## 2. 其他说明

- 此接口10.0.3 以上才支持

# ap.onBackgroundAudioPause(CALLBACK)

暂停播放音乐时触发此事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<pre id="J_output" class="output"></pre>
<script>
    var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
    var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
    var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
    var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
    var btnGetState = document.querySelector('#J_btnGetState');
    var output = document.querySelector('#J_output');
```

```
btnPlayBackgroundAudio.addEventListener('click', function(){
    ap.playBackgroundAudio({
        title: '舒缓轻音乐',
        url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflZgjBdxE.mp3',
        logo: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png',
        cover: 'https://zos.alipayobjects.com/rmsportal/SKMiZgNnNTQNdJWzQfdI.JPEG'
```

```
});
});
btnPauseBackgroundAudio.addEventListener('click', function(){
    ap.pauseBackgroundAudio();
});
btnStopBackgroundAudio.addEventListener('click', function(){
    ap.stopBackgroundAudio();
});
btnSeekBackgroundAudio.addEventListener('click', function(){
    ap.seekBackgroundAudio(0);
});
btnGetState.addEventListener('click', function(){
    ap.getBackgroundAudioPlayerState(function(res){
        output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
});
ap.onBackgroundAudioPlay(function(){
    ap.showToast('开始播放')
});
ap.onBackgroundAudioPause(function(){
    ap.showToast('暂停播放')
});
ap.onBackgroundAudioStop(function(){
    ap.showToast('停止播放')
});
</script>
```

## 2. 其他说明

- 此接口 10.0.3 以上才支持

# ap.onBackgroundAudioPlay(CALLBACK)

开始播放音乐时触发此事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
```

```
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<pre id="J_output" class="output"></pre>
<script>
    var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
    var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
    var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
    var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
    var btnGetState = document.querySelector('#J_btnGetState');
    var output = document.querySelector('#J_output');

    btnPlayBackgroundAudio.addEventListener('click', function(){
        ap.playBackgroundAudio({
            title: '舒缓轻音乐',
            url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflDZgjBdxE.mp3',
            logo: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png',
            cover: 'https://zos.alipayobjects.com/rmsportal/SKMizgNnNTQNdJWzQfdI.JPEG'
        });
    });
    btnPauseBackgroundAudio.addEventListener('click', function(){
        ap.pauseBackgroundAudio();
    });
    btnStopBackgroundAudio.addEventListener('click', function(){
        ap.stopBackgroundAudio();
    });
    btnSeekBackgroundAudio.addEventListener('click', function(){
        ap.seekBackgroundAudio(0);
    });
    btnGetState.addEventListener('click', function(){
        ap.getBackgroundAudioPlayerState(function(res){
            output.innerHTML = JSON.stringify(res, undefined, ' ');
        });
    });
    ap.onBackgroundAudioPlay(function(){
        ap.showToast('开始播放')
    });
    ap.onBackgroundAudioPause(function(){
        ap.showToast('暂停播放')
    });
    ap.onBackgroundAudioStop(function(){
        ap.showToast('停止播放')
    });
</script>
```

2. 其他说明

- 此接口 10.0.3 以上才支持

# ap.onBackgroundAudioStop(CALLBACK)

停止播放音乐时触发此事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<pre id="J_output" class="output"></pre>
<script>
  var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
  var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
  var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
  var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
  var btnGetState = document.querySelector('#J_btnGetState');
  var output = document.querySelector('#J_output');

  btnPlayBackgroundAudio.addEventListener('click', function(){
    ap.playBackgroundAudio({
      title: '舒缓轻音乐',
      url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflDZgjBdxE.mp3',
      logo: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png',
      cover: 'https://zos.alipayobjects.com/rmsportal/SKMIZgNnNTQNdJWzQfdI.JPEG'
    });
  });
  btnPauseBackgroundAudio.addEventListener('click', function(){
    ap.pauseBackgroundAudio();
  });
  btnStopBackgroundAudio.addEventListener('click', function(){
    ap.stopBackgroundAudio();
  });
  btnSeekBackgroundAudio.addEventListener('click', function(){
    ap.seekBackgroundAudio(0);
  });
</script>
```

```
btnGetState.addEventListener('click', function(){
    ap.getBackgroundAudioPlayerState(function(res){
        output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
});
ap.onBackgroundAudioPlay(function(){
    ap.showToast('开始播放')
});
ap.onBackgroundAudioPause(function(){
    ap.showToast('暂停播放')
});
ap.onBackgroundAudioStop(function(){
    ap.showToast('停止播放')
});
</script>
```

## 2. 其他说明

- 此接口 10.0.3 以上才支持

# ap.pauseBackgroundAudio(CALLBACK)

暂停播放音乐。

## 1. 代码示例



```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<pre id="J_output" class="output"></pre>
<script>
  var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
  var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
  var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
  var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
  var btnGetState = document.querySelector('#J_btnGetState');
  var output = document.querySelector('#J_output');

  btnPlayBackgroundAudio.addEventListener('click', function(){
    ap.playBackgroundAudio({
      url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflZgjBdxE.mp3',
      title: '舒缓轻音乐',
      cover: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png'
    });
  });
  btnPauseBackgroundAudio.addEventListener('click', function(){
    ap.pauseBackgroundAudio();
  });
  btnStopBackgroundAudio.addEventListener('click', function(){
    ap.stopBackgroundAudio();
  });
  btnSeekBackgroundAudio.addEventListener('click', function(){
    ap.seekBackgroundAudio(0);
  });
  btnGetState.addEventListener('click', function(){
    ap.getBackgroundAudioPlayerState(function(res){
      output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
  });
</script>
```

## 2. 其他说明

- 此接口1 0.0.3 以上才支持

# ap.playBackgroundAudio(OPTION, CALLBACK)

使用后台播放器播放音乐。

## 1. OPTION 参数说明

| 名称       | 类型     | 必选 | 描述       |
|----------|--------|----|----------|
| url      | String | 是  | 音乐链接地址   |
| title    | String | 否  | 音乐标题     |
| singer   | String | 否  | 演唱者      |
| describe | String | 否  | 音乐描述     |
| logo     | String | 否  | logo URL |
| cover    | String | 否  | 封面 URL   |

## 2. 错误码描述

| error | 描述        |
|-------|-----------|
| 12    | 失败时会有对应描述 |

## 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<pre id="J_output" class="output"></pre>
<script>
  var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
  var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
  var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
  var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
  var btnGetState = document.querySelector('#J_btnGetState');
  var output = document.querySelector('#J_output');

  btnPlayBackgroundAudio.addEventListener('click', function(){
    ap.playBackgroundAudio({
      url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflZgjBdxE.mp3',
      title: '舒缓轻音乐',
      cover: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png'
    });
  });
  btnPauseBackgroundAudio.addEventListener('click', function(){
    ap.pauseBackgroundAudio();
  });
  btnStopBackgroundAudio.addEventListener('click', function(){
    ap.stopBackgroundAudio();
  });
  btnSeekBackgroundAudio.addEventListener('click', function(){
    ap.seekBackgroundAudio(0);
  });
  btnGetState.addEventListener('click', function(){
    ap.getBackgroundAudioPlayerState(function(res){
      output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
  });
</script>
```

# ap.seekBackgroundAudio(OPTION | position, CALLBACK)

控制音乐播放进度。

# 1. OPTION 参数说明

| 名称       | 类型     | 必选 | 描述       |
|----------|--------|----|----------|
| position | Number | 是  | 音乐位置，单位秒 |

# 2. 错误码描述

| error | 描述        |
|-------|-----------|
| 12    | 失败时会有对应描述 |

# 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<pre id="J_output" class="output"></pre>
<script>
  var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
  var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
  var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
  var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
  var btnGetState = document.querySelector('#J_btnGetState');
  var output = document.querySelector('#J_output');

  btnPlayBackgroundAudio.addEventListener('click', function(){
    ap.playBackgroundAudio({
      url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflZgjBdxE.mp3',
      title: '舒缓轻音乐',
      cover: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png'
    });
  });
  btnPauseBackgroundAudio.addEventListener('click', function(){
    ap.pauseBackgroundAudio();
  });
  btnStopBackgroundAudio.addEventListener('click', function(){
    ap.stopBackgroundAudio();
  });
  btnSeekBackgroundAudio.addEventListener('click', function(){
    ap.seekBackgroundAudio(0);
  });
  btnGetState.addEventListener('click', function(){
    ap.getBackgroundAudioPlayerState(function(res){
      output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
  });
</script>
```

## 4. 其他说明

- 此接口1 0.0.3 以上才支持

# ap.stopBackgroundAudio(CALLBACK)

停止播放音乐。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btnPlayBackgroundAudio" class="btn btn-default">开始播放</button>
<button id="J_btnPauseBackgroundAudio" class="btn btn-default">暂停播放</button>
<button id="J_btnStopBackgroundAudio" class="btn btn-default">停止播放</button>
<button id="J_btnSeekBackgroundAudio" class="btn btn-default">从0开始播放</button>
<button id="J_btnGetState" class="btn btn-default">获取播放状态信息</button>
<pre id="J_output" class="output"></pre>
<script>
  var btnPlayBackgroundAudio = document.querySelector('#J_btnPlayBackgroundAudio');
  var btnPauseBackgroundAudio = document.querySelector('#J_btnPauseBackgroundAudio');
  var btnStopBackgroundAudio = document.querySelector('#J_btnStopBackgroundAudio');
  var btnSeekBackgroundAudio = document.querySelector('#J_btnSeekBackgroundAudio');
  var btnGetState = document.querySelector('#J_btnGetState');
  var output = document.querySelector('#J_output');

  btnPlayBackgroundAudio.addEventListener('click', function(){
    ap.playBackgroundAudio({
      url: 'https://os.alipayobjects.com/rmsportal/F0jtUAVjVsflDZgjBdxE.mp3',
      title: '舒缓轻音乐',
      cover: 'https://zos.alipayobjects.com/rmsportal/VRrMXbgUSdsibQjwZpyT.png'
    });
  });
  btnPauseBackgroundAudio.addEventListener('click', function(){
    ap.pauseBackgroundAudio();
  });
  btnStopBackgroundAudio.addEventListener('click', function(){
    ap.stopBackgroundAudio();
  });
  btnSeekBackgroundAudio.addEventListener('click', function(){
    ap.seekBackgroundAudio(0);
  });
  btnGetState.addEventListener('click', function(){
    ap.getBackgroundAudioPlayerState(function(res){
      output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
  });
</script>
```

## 2. 其他说明

- 此接口1 0.0.3 以上才支持

# ap.downloadFile(OPTION, CALLBACK)

上传本地资源到开发者服务器。

## 1. OPTION 参数说明

| 名称      | 类型     | 必选 | 描述             |
|---------|--------|----|----------------|
| url     | String | 是  | 下载文件地址         |
| headers | Object | 否  | HTTP 请求 Header |

## 2. CALLBACK 参数说明

| 名称         | 类型     | 描述        |
|------------|--------|-----------|
| apFilePath | String | 文件临时存放的位置 |

## 3. 错误码说明

| error | 描述   |
|-------|------|
| 12    | 下载失败 |
| 13    | 没有权限 |

## 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.img{ display:block; max-width: 100%; margin: 20px auto;}
</style>
<button id="J_btn" class="btn btn-default">下载图片</button>
<img id="J_img" class="img" />
<script>
  var btn = document.querySelector('#J_btn');
  var img = document.querySelector('#J_img');
  btn.addEventListener('click', function(){
    ap.showLoading('加载中...');
    ap.downloadFile({
      url: 'https://zos.alipayobjects.com/rmsportal/MMpnSrfxpnMhBKlECDDv.jpg'
    }, function(result){
      ap.hideLoading();
      img.src = result.apFilePath;
    });
  });
</script>
```

# ap.uploadFile(OPTION, CALLBACK)

上传本地资源到开发者服务器。

## 1. OPTION 参数说明

| 名称       | 类型     | 必选 | 描述  |
|----------|--------|----|---|
| url      | String | 是  | 开发者服务器地址                                    |
| filePath | String | 是  | 要上传文件资源的本地定位符                               |
| fileName | String | 是  | 文件名，即对应的 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 |
| fileType | String | 是  | 文件类型，image/video                            |
| headers  | Object | 否  | HTTP 请求 Header                              |
| formData | Object | 否  | HTTP 请求中其他额外的 form 数据                       |



## 2. CALLBACK 参数说明

| 名称         | 类型     | 描述            |
|------------|--------|---------------|
| data       | String | 服务器返回的数据      |
| statusCode | String | HTTP 状态码      |
| headers    | Object | 服务器返回的 header |

## 3. 错误码说明

| error | 描述     |
|-------|--------|
| 11    | 文件不存在  |
| 12    | 上传文件失败 |
| 13    | 没有权限   |

## 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn_image" class="btn btn-default">选择图片并上传</button>
```

```
<button id="J_btn_video" class="btn btn-default">选择视频并上传</button>
```

```
<script>
```

```
var btnImage = document.querySelector('#J_btn_image');
```

```
var btnVideo = document.querySelector('#J_btn_video');
```

```
btnImage.addEventListener('click', function(){
```

```
  ap.chooseImage(function(res){
```

```
    if(res.apFilePaths && res.apFilePaths.length > 0) {
```

```
      ap.uploadFile({
```

```
        url: '请使用自己服务器地址',
```

```
        fileType: 'image',
```

```
        fileName: 'file',
```

```
        filePath: res.apFilePaths[0]
```

```
      }, function(res){
```

```
        if (!res.hasOwnProperty('error')) {
```

```
          ap.alert('上传成功');
```

```
        }
```

```
      });
```

```
    }
```

```
  });
```

```
});
```

```
btnVideo.addEventListener('click', function(){
```

```
  ap.chooseVideo(3, function(res){
```

```
    if(res.apFilePath) {
```

```
      ap.uploadFile({
```

```
        url: '请使用自己服务器地址',
```

```
        fileType: 'video',
```

```
        fileName: 'file',
```

```
        filePath: res.apFilePath
```

```
      }, function(res){
```

```
        if (!res.hasOwnProperty('error')) {
```

```
          ap.alert('上传成功');
```

```
        }
```

```
      });
```

```
    }
```

```
  });
```

```
});
```

```
</script>
```

## ap.chooseImage(OPTION | count, CALLBACK)

拍照或从手机相册中选择图片。可直接传入一个数字作为 `OPTION.count` 参数。

## 1. OPTION 参数说明

| 名称         | 类型           | 必选 | 描述                             |
|------------|--------------|----|--------------------------------|
| count      | Number       | 否  | 最大可选照片数，默认1张，上限9张              |
| sourceType | String Array | 否  | 相册选取或者拍照，默认 ['camera','album'] |

## 2. CALLBACK 参数说明

| 名称          | 类型           | 描述     |
|-------------|--------------|--------|
| apFilePaths | String Array | 图片文件路径 |

## 3. 错误码描述

| error | 描述         |
|-------|------------|
| 10    | 用户取消操作     |
| 11    | 操作失败（权限不够） |

## 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>.img{ display:block; max-width: 100%; margin: 20px auto;}</style>
<button id="J_btn" class="btn btn-default">选择图片</button>
<img id="J_img" class="img" />
<script>
  var btn = document.querySelector('#J_btn');
  var img = document.querySelector('#J_img');
  btn.addEventListener('click', function(){
    ap.chooseImage(2, function(res){
      ap.alert(JSON.stringify(res));
      img.src = res.apFilePaths[0];
    });
  });
</script>
```

## 5. 其他说明

- 支付宝从 10.0.3 开始支持 apFilePath。

# ap.compressImage(OPTION | apFilePaths, CALLBACK)

压缩图片。可直接传入一个数组作为 OPTION. apFilePaths 参数。

## 1. OPTION 参数说明

| 名称          | 类型           | 必选 | 描述                                   |
|-------------|--------------|----|--------------------------------------|
| apFilePaths | String Array | 是  | 要压缩的图片文件路径数组                         |
| level       | Number       | 否  | 压缩级别，支持 0 ~ 4 的整数，默认 4。详见「level 说明表」 |

### level 说明

| level | 说明   |
|-------|------|
| 0     | 低质量  |
| 1     | 中等质量 |
|       |      |

|   |        |
|---|--------|
| 2 | 高质量    |
| 3 | 不压缩    |
| 4 | 根据网络适应 |

## 2. CALLBACK 参数说明

| 名称          | 类型           | 描述         |
|-------------|--------------|------------|
| apFilePaths | String Array | 压缩后的图片文件路径 |

## 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>.img{ display:block; max-width: 100%; margin: 20px auto;}</style>
<button id="J_btn" class="btn btn-default">选择图片</button>
<img id="J_img" class="img" />
<script>
  var btn = document.querySelector('#J_btn');
  var img = document.querySelector('#J_img');
  btn.addEventListener('click', function(){
    ap.chooseImage(2, function(res){
      ap.compressImage({
        apFilePaths: res.apFilePaths,
        level: 0
      }, function(result){
        img.src = result.apFilePaths[0]
      });
    });
  });
</script>
```

## 4. 其他说明

- 支付宝 10.0.12 开始支持此接口。

# ap.previewImage(OPTION | urls, CALLBACK)

预览图片。可直接传入一个图片链接数组作为 OPTION.urls 参数。

## 1. OPTION 参数说明

| 名称      | 类型     | 必选 | 描述            |
|---------|--------|----|---------------|
| urls    | Arrary | 是  | 要预览的图片链接列表    |
| current | Number | 否  | 当前显示图片索引，默认 0 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_preview" class="btn btn-default">预览图片</button>
<script>
  var btnPreview = document.querySelector('#J_btn_preview');
  btnPreview.addEventListener('click', function(){
    ap.previewImage({
      current: 2,
      urls: [
        'https://img.alicdn.com/tps/TB1sXGYIFXXXXc5XpXXXXXXXXXX.jpg',
        'https://img.alicdn.com/tps/TB1pfG4IFXXXXc6XXXXXXXXXXXXX.jpg',
        'https://img.alicdn.com/tps/TB1h9xxIFXXXXbKXXXXXXXXXXXXX.jpg'
      ]
    });
  });
</script>
```

# ap.saveImage(OPTION | url, CALLBACK)

保存在线图片到手机相册。可直接传入一个图片链接地址作为 OPTION.url 参数。

## 1. OPTION 参数说明

| 名称              | 类型      | 必选 | 描述                 |
|-----------------|---------|----|--------------------|
| url             | String  | 是  | 要保存的图片链接           |
| showActionSheet | Boolean | 否  | 是否显示图片操作菜单，默认 true |

## 2. CALLBACK 参数说明

**注意：** 如果接口在调用时传了回调函数，那么保存结果交给此回调处理，接口默认不再弹 toast 提示用户。

## 3. 错误码说明

| error | 描述                   |
|-------|----------------------|
| 2     | 参数无效，没有传 url 参数      |
| 15    | 没有开启相册权限(ios only)   |
| 16    | 手机相册存储空间不足(ios only) |
| 17    | 保存图片过程中的其他错误         |

## 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_save" class="btn btn-default">保存图片</button>
<script>
  var btnSave = document.querySelector('#J_btn_save');
  btnSave.addEventListener('click', function(){
    ap.saveImage('https://img.alicdn.com/tps/TB1sXGYIFXXXc5XpXXXXXXXXXX.jpg');
  });
</script>
```

# ap.chooseVideo(OPTION | maxDuration, CALLBACK)

录制或从手机相册中选择视频。可直接传入一个数字作为 OPTION.maxDuration 参数。

## 1. OPTION 参数说明

| 名称          | 类型           | 必选 | 描述                                  |
|-------------|--------------|----|-------------------------------------|
| maxDuration | Number       | 否  | 最大录制时长，单位秒，默认60s                    |
| sourceType  | String Array | 否  | 相册选取或者拍照，默认 ['camera','album']      |
| camera      | String Array | 否  | 前置或者后置摄像头，默认前后都有，即：[‘front’，‘back’] |

## 2. CALLBACK 参数说明

| 名称         | 类型     | 描述            |
|------------|--------|---------------|
| apFilePath | String | 图片文件路径        |
| duration   | Number | 选定视频的时间长度，单位秒 |
| size       | Number | 选定视频的数据量大小    |
| width      | Number | 选定视频的宽        |
| height     | Number | 选定视频的高        |

## 3. 错误码描述

| error | 描述         |
|-------|------------|
| 10    | 用户取消操作     |
| 11    | 操作失败（权限不够） |
| 12    | 数据处理失败     |

## 4. 代码示例



```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn" class="btn btn-default">选择视频</button>
```

```
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.chooseVideo(3, function(res){
      ap.alert(JSON.stringify(res));
    });
  });
</script>
```

# ap.offAppPause(CALLBACK)

停止监听应用压后台事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<h2 class="am-ft-center">点击 Home 键回到手机桌面</h2>
```

```
<script>
  var output = document.querySelector('#J_output');
  var appPauseHandler = function(){
    time = Date.now();
    ap.offAppPause(appPauseHandler);
  };
  var appResumeHandler = function(){
    time = Date.now() - time;
    ap.alert('应用在后台运行了' + time/1000 + 's');
    ap.offAppPause(appResumeHandler);
  };
  var time;

  ap.onAppPause(appPauseHandler);
  ap.onAppResume(appResumeHandler);
</script>
```

# ap.offAppResume(CALLBACK)

停止监听应用压后台事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2 class="am-ft-center">点击 Home 键回到手机桌面</h2>
<script>
  var output = document.querySelector('#J_output');
  var appPauseHandler = function(){
    time = Date.now();
    ap.offAppPause(appPauseHandler);
  };
  var appResumeHandler = function(){
    time = Date.now() - time;
    ap.alert('应用在后台运行了' + time/1000 + 's');
    ap.offAppResume(appResumeHandler);
  };
  var time;

  ap.onAppPause(appPauseHandler);
  ap.onAppResume(appResumeHandler);
</script>
```

# ap.onAppPause(CALLBACK)

当应用压后台时触发。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<h2 class="am-ft-center">点击 Home 键回到手机桌面</h2>
```

```
<script>
  var output = document.querySelector('#J_output');
  var time;
  ap.onPause(function(){
    time = Date.now();
  });
  ap.onResume(function(){
    time = Date.now() - time;
    ap.alert('应用在后台运行了' + time/1000 + 's');
  });
</script>
```

# ap.onResume(CALLBACK)

当应用从后台唤起时触发。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<h2 class="am-ft-center">点击 Home 键回到手机桌面</h2>
```

```
<script>
  var output = document.querySelector('#J_output');
  var time;
  ap.onPause(function(){
    time = Date.now();
  });
  ap.onResume(function(){
    time = Date.now() - time;
    ap.alert('应用在后台运行' + time/1000 + 's');
  });
</script>
```

# ap.offPause(CALLBACK)

停止监听 pause 事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn_push" class="btn btn-default">打开新页面</button>
```

```
<script>
```

```
  var btnPush = document.querySelector('#J_btn_push');
```

```
  var handler = function(){
```

```
    ap.alert('亲，前面那个页面 pause 了!');
```

```
    ap.offBack(handler);
```

```
  };
```

```
  btnPush.addEventListener('click', function(){
```

```
    ap.pushWindow('onResume-demo.html?for=resume');
```

```
  });
```

```
  ap.onPause(handler);
```

```
</script>
```

# ap.offResume(CALLBACK)

停止监听 resume 事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_push" class="btn btn-default">打开新页面</button>
<script>
  var btnPush = document.querySelector('#J_btn_push');
  var handler = function(res){
    ap.alert('欢迎加来');
    ap.offResume(handler);
  };
  ap.onResume(handler);
  btnPush.addEventListener('click', function(){
    ap.pushWindow({
      url: 'onResume-demo.html',
      data: {
        for: 'resume'
      }
    });
  });
}</script>
```

# ap.onPause(CALLBACK)

当一个页面不可见时(被压入后台、锁屏、pushWindow 到下个页面)，会触发此事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_push" class="btn btn-default">打开新页面</button>
<script>
  var btnPush = document.querySelector('#J_btn_push');
  btnPush.addEventListener('click', function(){
    ap.pushWindow('onResume-demo.html?for=resume');
  });
  ap.onPause(function(){
    ap.alert('亲，前面那个页面 pause 了!');
  });
}</script>
```

# ap.onResume(CALLBACK)

当一个页面重新回到栈顶时(从后台被唤起、锁屏界面恢复、从下个页面回退)，会触发此事件。  
如果这个页面是通过 popWindow/popTo 到达，且传递了 data 参数，此页可以获取到 data。

## 1. CALLBACK 参数说明

| 名称   | 类型     | 描述                             |
|------|--------|--------------------------------|
| data | Object | 通过 popWindow/popTo 传递的 data 参数 |

## 2. 代码示例

<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J\_btn\_pop" style="display:none" class="btn btn-default">退出页面(带数据)</button>  
<button id="J\_btn\_push" style="display:none" class="btn btn-default">打开新页面</button>

```
<script>
  var btnPush = document.querySelector('#J_btn_push');
  var btnPop = document.querySelector('#J_btn_pop');
  if (ap.parseQueryString().for === 'resume') {
    btnPop.style.display = 'block'
  } else {
    btnPush.style.display = 'block';

    ap.onResume(function(res){
      ap.alert('城市: ' + res.data.city);
    });
  }
  btnPush.addEventListener('click', function(){
    ap.pushWindow({
      url: 'onResume-demo.html',
      data: {
        for: 'resume'
      }
    });
  });
  btnPop.addEventListener('click', function(){
    ap.popWindow({
      data: {
        city: '上海',
        adCode: '310100'
      }
    });
  });
}</script>
```

# ap.popTo(OPTION | urlPattern | index, CALLBACK)

popTo 可以一次回退多级页面。可直接传入一个字符串作为 OPTION.urlPattern 参数，或直接传入一个整数作为 OPTION.index 参数。

## 1. OPTION 参数说明

| 参数 | 类型 | 必填 | 描述 |
|----|----|----|----|
|    |    |    |    |

|            |        |   |   |
|------------|--------|---|---|
| url        | String | 否 | 目标页面的完整 URL                               |
| urlPattern | String | 否 | 目标页面的 URL 匹配表达式（URL 如果包含 urlPattern，匹配成功） |
| index      | Number | 否 | 目标页面在会话页面栈中的索引；如果小于零，则将与当前页面的 index 相加    |
| data       | Object | 否 | 传递的 data 对象将会被即将露出的页面通过 onResume 事件接收     |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_pop" class="btn btn-default">退出页面</button>
<script>
  var btnPop = document.querySelector('#J_btn_pop');
  btnPop.addEventListener('click', function(){
    ap.popTo(-1);
  });
</script>
```

## 3. 其他说明

- 一般情况下，popTo 用于分多步才能完成的场景，比如三级地址选择后返回等。
- 如果通过 url 或者 urlPattern 来返回，popTo 会返回到离当前页面最远的页面，也就是栈底。同时不会去检测当前页面的 url 是否符合。
- popTo 时所带的 data 如何被接收？ 请查看onResume (../event/onResume.html)

# ap.popWindow(OPTION | data)

关闭当前页面。可直接传入一个对象作为 OPTION.data 参数。

## 1. OPTION 参数说明

| 参数   | 类型     | 必填 | 描述                                    |
|------|--------|----|---------------------------------------|
| data | Object | 否  | 传递的 data 对象将会被即将露出的页面通过 onResume 事件接收 |



## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_pop" class="btn btn-default">退出页面</button>
<script>
  var btnPop = document.querySelector('#J_btn_pop');
  btnPop.addEventListener('click', function(){
    ap.popWindow();
  });
</script>
```

## 3. 其他说明

- popWindow 时所带的数据如何被接收？ 请查看onResume (../event/onResume.html)

# ap.pushWindow(OPTION | url)

pushWindow用来打开一个新的页面，自带转场动画。可直接传入一个字符串作为 OPTION.url 参数。

## 1. OPTION 参数说明

| 参数   | 类型     | 必填 | 描述   |
|------|--------|----|--|
| url  | String | 是  | 要打开的页面url  |
| data | Object | 否  | url的参数，会以 query string 跟在 url 后面。在打开的新页面里可以用 ap.parseQueryString() (../data/querystring/parseQueryStringSync.html) 来获取 |
|      |        |    |  |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn_url" class="btn btn-default">打开淘宝首页</button>
<button id="J_btn_withData" class="btn btn-default">打开高德地图搜索页面</button>
<script>
  var btnUrl = document.querySelector('#J_btn_url');
  var btnWithData = document.querySelector('#J_btn_withData');
  btnUrl.addEventListener('click', function(){
    ap.pushWindow('https://m.taobao.com/#index');
  });
  btnWithData.addEventListener('click', function(){
    ap.pushWindow({
      url: 'http://uri.amap.com/search',
      data: {
        keyword: '美食',
        city: '310100',
        view: 'map'
      }
    });
  });
</script>
```

### 3. 其他说明

- `ap.pushWindow` 与 `location.href` 有所区别，类同于PC浏览器的新开标签页，每个window都是一个新的标签页，因此原页面仅仅是被压到后台，状态始终保持，JS 也会继续运行。

# ap.redirectTo(OPTION | url)

替换当前页面，并且不会产生历史记录。可直接传入一个字符串作为 `OPTION.url` 参数。

## 1. OPTION 参数说明

| 参数   | 类型     | 必填 | 描述   |
|------|--------|----|--|
| url  | String | 是  | 要打开的页面url  |
| data | Object | 否  | url的参数，会以 query string 跟在 url 后面。在打开的新页面里可以用 <code>ap.parseQueryString()</code> ( <code>../data/parseQueryString.html</code> ) 来获取 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_url" class="btn btn-default">打开天猫首页</button>
<button id="J_btn_withData" class="btn btn-default">打开高德地图搜索页面</button>
<script>
  var btnUrl = document.querySelector('#J_btn_url');
  var btnWithData = document.querySelector('#J_btn_withData');
  btnUrl.addEventListener('click', function(){
    ap.redirectTo('https://m.tmall.com');
  });
  btnWithData.addEventListener('click', function(){
    ap.redirectTo({
      url: 'http://uri.amap.com/search',
      data: {
        keyword: '支付宝',
        city: '310100',
        view: 'map'
      }
    });
  });
</script>
```

# ap.closeSocket()

关闭WebSocket连接。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
.img{ display:block; max-width: 100%; margin: 20px auto;}
</style>
<button id="J_btn_connect" class="btn btn-default">建立链接</button>
<button id="J_btn_send" class="btn btn-default">发送数据</button>
<button id="J_btn_close" class="btn btn-default">关闭链接</button>

<script>
```

```
var btnConnect = document.querySelector('#J_btn_connect');
var btnSend = document.querySelector('#J_btn_send');
var btnClose = document.querySelector('#J_btn_close');
var counter = 0;
ap.onSocketOpen(function(){
    ap.showToast('socket 已开启');
});
ap.onSocketError(function(){
    ap.showToast('socket 出错');
});
ap.onSocketClose(function(){
    ap.showToast('socket 已关闭');
});
ap.onSocketMessage(function(res) {
    ap.showToast('收到服务器内容: ' + res.data)
})
btnConnect.addEventListener('click', function(){
    ap.connectSocket({
        url: 'ws://127.0.0.1:8088',
        data: {
            b: '1234567890',
            c: 9999
        },
        headers: {
            a: '0987654321'
        }
    });
});
btnSend.addEventListener('click', function(){
    counter ++;
    ap.sendSocketMessage({
        data: counter+' '
    }, function(res){
        console.warn(res);
    });
});
btnClose.addEventListener('click', function(){
    counter ++;
    ap.closeSocket();
});
```

</script>

# ap.connectSocket(OPTION | url, CALLBACK)

向指定服务器发起 WebSocket 请求。可直接传入一个字符串作为 OPTION.url 参数。

## 1. OPTION 参数说明

| 名称      | 类型     | 必选 | 描述        |
|---------|--------|----|-----------|
| url     | String | 是  | 目标服务器 url |
| data    | Object | 否  | 请求的参数     |
| headers | Object | 否  | 设置请求的头部信息 |

## 2. 错误码说明

| error | 描述   |
|-------|--|
| 1     | Error Unknown.   |
| 2     | Connect when already connected.                                |
| 3     | Empty URL.   |
| 4     | Illegal URL format.  |
| 5     | URL scheme not ws or wss                                       |
| 6     | Timed out connecting to server.                                |
| 7     | Invalid server certificate.                                    |
| 8     | Invalid Sec-WebSocket-Accept response.                         |
| 9     | Server specified Sec-WebSocket-Protocol that wasn't requested. |
| 10    | Invalid State: Cannot send until connection is open.           |
| 11    | Error writing to stream.                                       |
| 12    | Unable to allocate memory to read from socket.                 |

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
.img{ display:block; max-width: 100%; margin: 20px auto;}
</style>
<button id="J_btn_connect" class="btn btn-default">建立链接</button>
<button id="J_btn_send" class="btn btn-default">发送数据</button>
<button id="J_btn_close" class="btn btn-default">关闭链接</button>

<script>
  var btnConnect = document.querySelector('#J_btn_connect');
  var btnSend = document.querySelector('#J_btn_send');
  var btnClose = document.querySelector('#J_btn_close');
  var counter = 0;
  ap.onSocketOpen(function(){
    ap.showToast('socket 已开启');
  });
  ap.onSocketError(function(){
    ap.showToast('socket 出错');
  });
  ap.onSocketClose(function(){
    ap.showToast('socket 已关闭');
  });
  ap.onSocketMessage(function(res) {
    ap.showToast('收到服务器内容: ' + res.data)
  })
  btnConnect.addEventListener('click', function(){
    ap.connectSocket({
      url: 'ws://127.0.0.1:8088',
      data: {
        b: '1234567890',
        c: 9999
      },
      headers: {
        a: '0987654321'
      }
    });
  });
  btnSend.addEventListener('click', function(){
    counter ++;
    ap.sendSocketMessage({
      data: counter+' '
    }, function(res){
      console.warn(res);
    });
  });
});
```

```
    btnClose.addEventListener('click', function(){
        counter ++;
        ap.closeSocket();
    });
```

```
</script>
```

# ap.offSocketClose()

移除 WebSocket 关闭事件监听。

# ap.offSocketError()

移除 WebSocket 报错事件监听。

# ap.offSocketMessage()

移除 WebSocket 消息事件监听。

# ap.offSocketOpen()

移除 WebSocket 打开事件监听。

# ap.onSocketClose(CALLBACK)

监听 WebSocket 关闭事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
.img{ display:block; max-width: 100%; margin: 20px auto;}
</style>
<button id="J_btn_connect" class="btn btn-default">建立链接</button>
<button id="J_btn_send" class="btn btn-default">发送数据</button>
<button id="J_btn_close" class="btn btn-default">关闭链接</button>

<script>
  var btnConnect = document.querySelector('#J_btn_connect');
  var btnSend = document.querySelector('#J_btn_send');
  var btnClose = document.querySelector('#J_btn_close');
  var counter = 0;
  ap.onSocketOpen(function(){
    ap.showToast('socket 已开启');
  });
  ap.onSocketError(function(){
    ap.showToast('socket 出错');
  });
  ap.onSocketClose(function(){
    ap.showToast('socket 已关闭');
  });
  ap.onSocketMessage(function(res) {
    ap.showToast('收到服务器内容: ' + res.data)
  })
  btnConnect.addEventListener('click', function(){
    ap.connectSocket({
      url: 'ws://127.0.0.1:8088',
      data: {
        b: '1234567890',
        c: 9999
      },
      headers: {
        a: '0987654321'
      }
    });
```



```
});
btnSend.addEventListener('click', function(){
    counter ++;
    ap.sendMessage({
        data: counter+' '
    }, function(res){
        console.warn(res);
    });
});
btnClose.addEventListener('click', function(){
    counter ++;
    ap.closeSocket();
});
```

</script>

# ap.onSocketError(CALLBACK)

监听 WebSocket 错误事件。

## 1. 代码示例

<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

```
<style>
.output{ display:block; max-width: 100%; overflow: auto}
.img{ display:block; max-width: 100%; margin: 20px auto;}
</style>
<button id="J_btn_connect" class="btn btn-default">建立链接</button>
<button id="J_btn_send" class="btn btn-default">发送数据</button>
<button id="J_btn_close" class="btn btn-default">关闭链接</button>
```

```
<script>
    var btnConnect = document.querySelector('#J_btn_connect');
    var btnSend = document.querySelector('#J_btn_send');
    var btnClose = document.querySelector('#J_btn_close');
    var counter = 0;
    ap.onSocketOpen(function(){
        ap.showToast('socket 已开启');
    });
    ap.onSocketError(function(){
```

```
ap.showToast('socket 出错');
});
ap.onSocketClose(function(){
    ap.showToast('socket 已关闭');
});
ap.onSocketMessage(function(res) {
    ap.showToast('收到服务器内容: ' + res.data)
})
btnConnect.addEventListener('click', function(){
    ap.connectSocket({
        url: 'ws://127.0.0.1:8088',
        data: {
            b: '1234567890',
            c: 9999
        },
        headers: {
            a: '0987654321'
        }
    });
});
btnSend.addEventListener('click', function(){
    counter ++;
    ap.sendSocketMessage({
        data: counter+' '
    }, function(res){
        console.warn(res);
    });
});
btnClose.addEventListener('click', function(){
    counter ++;
    ap.closeSocket();
});
```

</script>

# ap.onSocketMessage(CALLBACK)

监听 WebSocket 接受到服务器的消息事件。

## 1. CALLBACK 参数说明

| 名称 | 类型 | 描述 |
|----|----|----|
|----|----|----|

|      |                    |          |
|------|--------------------|----------|
| data | String/ArrayBuffer | 服务器返回的消息 |
|------|--------------------|----------|

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
.img{ display:block; max-width: 100%; margin: 20px auto;}
</style>
<button id="J_btn_connect" class="btn btn-default">建立链接</button>
<button id="J_btn_send" class="btn btn-default">发送数据</button>
<button id="J_btn_close" class="btn btn-default">关闭链接</button>
<script>
  var btnConnect = document.querySelector('#J_btn_connect');
  var btnSend = document.querySelector('#J_btn_send');
  var btnClose = document.querySelector('#J_btn_close');
  var counter = 0;
  ap.onSocketOpen(function(){
    ap.showToast('socket 已开启');
  });
  ap.onSocketError(function(){
    ap.showToast('socket 出错');
  });
  ap.onSocketClose(function(){
    ap.showToast('socket 已关闭');
  });
  ap.onSocketMessage(function(res) {
    ap.showToast('收到服务器内容: ' + res.data)
  })
  btnConnect.addEventListener('click', function(){
    ap.connectSocket({
      url: 'ws://127.0.0.1:8088',
      data: {
        b: '1234567890',
        c: 9999
      },
      headers: {
        a: '0987654321'
      }
    });
  });
  btnSend.addEventListener('click', function(){
    counter ++;
    ap.sendSocketMessage({
      data: counter+' '
    });
  });
</script>
```

```
    }, function(res){
      console.warn(res);
    });
  });
  btnClose.addEventListener('click', function(){
    counter ++;
    ap.closeSocket();
  });
```

</script>

# ap.onSocketOpen(CALLBACK)

监听 WebSocket 连接打开事件。

## 1. 代码示例

<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

```
<style>
.output{ display:block; max-width: 100%; overflow: auto}
.img{ display:block; max-width: 100%; margin: 20px auto;}
</style>
<button id="J_btn_connect" class="btn btn-default">建立链接</button>
<button id="J_btn_send" class="btn btn-default">发送数据</button>
<button id="J_btn_close" class="btn btn-default">关闭链接</button>
```

```
<script>
  var btnConnect = document.querySelector('#J_btn_connect');
  var btnSend = document.querySelector('#J_btn_send');
  var btnClose = document.querySelector('#J_btn_close');
  var counter = 0;
  ap.onSocketOpen(function(){
    ap.showToast('socket 已开启');
  });
  ap.onSocketError(function(){
    ap.showToast('socket 出错');
  });
  ap.onSocketClose(function(){
    ap.showToast('socket 已关闭');
  });
```

```
ap.onSocketMessage(function(res) {
  ap.showToast('收到服务器内容: ' + res.data)
})
btnConnect.addEventListener('click', function(){
  ap.connectSocket({
    url: 'ws://127.0.0.1:8088',
    data: {
      b: '1234567890',
      c: 9999
    },
    headers: {
      a: '0987654321'
    }
  });
});
btnSend.addEventListener('click', function(){
  counter++;
  ap.sendMessage({
    data: counter+' '
  }, function(res){
    console.warn(res);
  });
});
btnClose.addEventListener('click', function(){
  counter++;
  ap.closeSocket();
});

</script>
```

# ap.sendMessage(OPTION, CALLBACK)

通过 WebSocket 连接发送数据，需要先 ap.connectSocket，并在 ap.onSocketOpen 回调之后才能发送。

## 1. OPTION 参数说明

| 名称   | 类型                 | 必选 | 描述    |
|------|--------------------|----|-------|
| data | String/ArrayBuffer | 否  | 请求的参数 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<style>
.output{ display:block; max-width: 100%; overflow: auto}
.img{ display:block; max-width: 100%; margin: 20px auto;}
</style>
<button id="J_btn_connect" class="btn btn-default">建立链接</button>
<button id="J_btn_send" class="btn btn-default">发送数据</button>
<button id="J_btn_close" class="btn btn-default">关闭链接</button>

<script>
  var btnConnect = document.querySelector('#J_btn_connect');
  var btnSend = document.querySelector('#J_btn_send');
  var btnClose = document.querySelector('#J_btn_close');
  var counter = 0;
  ap.onSocketOpen(function(){
    ap.showToast('socket 已开启');
  });
  ap.onSocketError(function(){
    ap.showToast('socket 出错');
  });
  ap.onSocketClose(function(){
    ap.showToast('socket 已关闭');
  });
  ap.onSocketMessage(function(res) {
    ap.showToast('收到服务器内容: ' + res.data)
  })
  btnConnect.addEventListener('click', function(){
    ap.connectSocket({
      url: 'ws://127.0.0.1:8088',
      data: {
        b: '1234567890',
        c: 9999
      },
      headers: {
        a: '0987654321'
      }
    });
  });
  btnSend.addEventListener('click', function(){
    counter ++;
    ap.sendSocketMessage({
      data: counter+' '
    }, function(res){
      console.warn(res);
    });
  });
});
```

```
btnClose.addEventListener('click', function(){
    counter ++;
    ap.closeSocket();
});

</script>
```

# ap.chooseCity(OPTION, CALLBACK)

打开城市选择列表。

## 1. OPTION 参数说明

| 名称              | 类型           | 必填 | 描述                   |
|-----------------|--------------|----|----------------------|
| showLocatedCity | Boolean      | 否  | 是否显示当前定位城市，默认 false  |
| showHotCities   | Boolean      | 否  | 是否显示热门城市，默认 true     |
| cities          | Object Array | 否  | 自定义城市列表，列表内对象字段见下表   |
| hotCities       | Object array | 否  | 自定义热门城市列表，列表内对象字段见下表 |

城市列表对象字段说明：

| 名称     | 类型     | 必填 | 描述               |
|--------|--------|----|------------------|
| city   | String | 是  | 城市名              |
| adCode | String | 是  | 行政区划代码           |
| spell  | String | 否  | 城市名对应拼音拼写，方便用户搜索 |

## 2. CALLBACK 参数说明

| 名称   | 类型     | 描述  |
|------|--------|-----|
| city | String | 城市名 |
|      |        |     |

|        |        |        |
|--------|--------|--------|
| adCode | String | 行政区划代码 |
|--------|--------|--------|

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btnGPS" class="btn btn-default">选择城市（带定位） </button>
<button id="J_btnCustom" class="btn btn-default">选择城市（自定义） </button>
```

```
<script>
  var btnGPS = document.querySelector('#J_btnGPS');
  var btnCustom = document.querySelector('#J_btnCustom');
  btnGPS.addEventListener('click', function(){
    ap.chooseCity({
      showLocatedCity: true
    }, function(res) {
      ap.alert(res.city + ':' + res.adCode);
    });
  });
  btnCustom.addEventListener('click', function(){
    ap.chooseCity({
      cities: [
        {
          city: '朝阳区',
          adCode: '110105',
          spell: 'chaoyang'
        },
        {
          city: '海淀区',
          adCode: '110108',
          spell: 'haidian'
        },
        {
          city: '丰台区',
          adCode: '110106',
          spell: 'fengtai'
        },
        {
          city: '东城区',
          adCode: '110101',
          spell: 'dongcheng'
        },
        {
          city: '西城区',
          adCode: '110102',
          spell: 'xicheng'
        },
        {

```



```
        city: '房山区',
        adCode: '110111',
        spell: 'fangshan'
      }
    ],
    hotCities: [
      {
        city: '朝阳区',
        adCode: '110105'
      },
      {
        city: '海淀区',
        adCode: '110108'
      },
      {
        city: '丰台区',
        adCode: '110106'
      }
    ]
  }, function(res){
    ap.alert(res.city + ':' + res.adCode);
  });
})
</script>
```

## 4. 其他说明

- 如果用户没有选择任何城市直接点击了返回，将不会触发回调函数。
- 最新县及县以上行政区划代码 (<http://www.stats.gov.cn/tjsj/tjbz/xzqhdm/>)

# ap.chooseAlipayContact(OPTION | count, CALLBACK)

唤起支付宝通讯录，选择一个或者多个支付宝联系人。

## 1. OPTION 参数说明

| 名称    | 类型     | 必填 | 描述                     |
|-------|--------|----|------------------------|
| count | Number | 否  | 单次最多选择联系人个数，默认 1，最大 10 |

## 2. CALLBACK 参数说明

| 名称       | 类型           | 描述                      |
|----------|--------------|-------------------------|
| contacts | Object Array | 选中的支付宝联系人数组，数组内部对象字段见下表 |

### contacts 返回字段说明

| 名称       | 类型     | 描述           |
|----------|--------|--------------|
| realName | String | 账号的真实姓名      |
| mobile   | String | 账号对应的手机号码    |
| email    | String | 账号的邮箱        |
| avatar   | String | 账号的头像链接      |
| userId   | String | 支付宝账号 userId |

## 3. 错误说明

| error | 描述     |
|-------|--------|
| 10    | 没有权限   |
| 11    | 用户取消操作 |

## 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<style>.output{ display:block; max-width: 100%; overflow: auto}</style>
<button id="J_btn" class="btn btn-default">选择支付宝联系人</button>
<pre id="J_output" class="output"></pre>
<script>
  var btn = document.querySelector('#J_btn');
  var output = document.querySelector('#J_output');
  btn.addEventListener('click', function(){
    ap.chooseAlipayContact(2, function(res){
      output.innerHTML = JSON.stringify(res, undefined, '  ');
    });
  });
</script>
```

## 5. 其他说明

- 返回的 mobile 和 email 字段不一定全有值，取决于所选取联系人的支付宝账号类型是手机号还是邮箱。

# ap.choosePhoneContact(CALLBACK)

选择本地系统通信录中某个联系人的电话。

## 1. CALLBACK 参数说明

| 名称     | 类型     | 描述        |
|--------|--------|-----------|
| name   | String | 选中的联系人姓名  |
| mobile | String | 选中的联系人手机号 |

## 2. 错误说明

| error | 描述                  |
|-------|---------------------|
| 10    | 没有权限；               |
| 11    | 用户取消操作(或设备未授权使用通讯录) |

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">选择联系人电话</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.choosePhoneContact(function(res){
      ap.alert('姓名: ' + res.name + '\n号码: ' + res.mobile);
    });
  });
</script>
```

## ap.datePicker(OPTION | formate, CALLBACK)

设置导航栏文字及样式。可直接传入一个字符串作为 OPTION.title 参数。

### 1. OPTION 参数说明

| 名称          | 类型     | 必填 | 描述  |
|-------------|--------|----|---|
| formate     | String | 否  | 返回的日期格式，默认 yyyy-MM-dd。支持 HH:mm:ss，yyyy-MM-dd，yyyy-MM-dd HH:mm:ss 三种格式 |
| currentDate | String | 否  | 初始选择的日期时间，默认当前时间  |
| startDate   | String | 否  | 最小日期时间  |
| endDate     | String | 否  | 最大日期时间  |

### 2. CALLBACK 参数说明

| 名称   | 类型     | 描述      |
|------|--------|---------|
| date | String | 选择的日期时间 |

### 3. 错误码说明

| error | 描述     |
|-------|--------|
| 11    | 用户取消操作 |

### 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btnDate" class="btn btn-default">选择日期</button>
<button id="J_btnTime" class="btn btn-default">选择时间</button>
<button id="J_btnDateTime" class="btn btn-default">选择日期时间</button>
<script>
  var btnDate = document.querySelector('#J_btnDate');
  var btnTime = document.querySelector('#J_btnTime');
  var btnDateTime = document.querySelector('#J_btnDateTime');
  btnDate.addEventListener('click', function(){
    ap.datePicker({
      formate: 'yyyy-MM-dd',
      currentDate: '2012-12-12',
      startDate: '2012-12-10',
      endDate: '2012-12-15'
    }, function(res) {
      ap.alert(res.date);
    });
  });
  btnTime.addEventListener('click', function(){
    ap.datePicker({
      formate: 'HH:mm:ss',
      currentDate: '12:12:12',
      startDate: '11:11:11',
      endDate: '13:13:13'
    }, function(res) {
      ap.alert(res.date);
    });
  });
  btnDateTime.addEventListener('click', function(){
    ap.datePicker({
      formate: 'yyyy-MM-dd HH:mm:ss',
      currentDate: '2012-01-09 11:11:11',
      startDate: '2012-01-01 11:11:11',
      endDate: '2012-01-10 11:11:11'
    }, function(res) {
      ap.alert(res.date);
    });
  });
}</script>
```

# ap.hideNavigationBarLoading()

隐藏导航栏 loading。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_show" class="btn btn-default">showNavigationBarLoading</button>
<button id="J_btn_hide" class="btn btn-default">hideNavigationBarLoading</button>
<script>
  var btnShow = document.querySelector('#J_btn_show');
  var btnHide = document.querySelector('#J_btn_hide');
  btnShow.addEventListener('click', function(){
    ap.showNavigationBarLoading();
  });
  btnHide.addEventListener('click', function(){
    ap.hideNavigationBarLoading();
  });
</script>
```

# ap.offTitleClick(CALLBACK)

停止监听 titleClick事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2 class="am-ft-center">点击导航栏文字</h2>
<script>
  var handler = function(){
    ap.alert('title 只能点一次');
    ap.offTitleClick(handler);
  };
  ap.onTitleClick(handler);
</script>
```

# ap.onTitleClick(CALLBACK)

点击导航栏文字时触发回调

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2 class="am-ft-center">点击导航栏文字</h2>
<script>
  ap.onTitleClick(function(){
    ap.alert('亲!');
  });
</script>
```

## 2. 其他说明

- onTitleClick 在点击文字时才触发，点击导航栏其他空白地方不会触发

# ap.setNavigationBar(OPTION | title, CALLBACK)

设置导航栏文字及样式。可直接传入一个字符串作为 OPTION.title 参数。

## 1. OPTION 参数说明

| 名称                | 类型      | 必填 | 描述   |
|-------------------|---------|----|--|
| title             | String  | 否  | 导航栏标题  |
| image             | String  | 否  | 图片链接地址，必须 https，请使用一张3x高清图。若设置了 image，则 title 参数失效   |
| backgroundColor   | String  | 否  | 导航栏背景色，支持16进制颜色值   |
| borderBottomColor | String  | 否  | 导航栏底部边框颜色，支持16进制颜色值。若设置了 backgroundColor，borderBottomColor 会不生效，默认会和 backgroundColor 颜色一样。 |
| reset             | Boolean | 否  | 是否重置导航栏为支付宝默认配色，默认 false。  |



## 2. CALLBACK 参数说明

设置后调用

## 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn_title" class="btn btn-default">设置标题文字</button>
<button id="J_btn_image" class="btn btn-default">设置标题图片(url)</button>
<button id="J_btn_bgColor" class="btn btn-default">设置导航栏背景色</button>
<button id="J_btn_borderColor" class="btn btn-default">设置导航栏颜底边框色</button>
<button id="J_btn_reset" class="btn btn-default">重置导航栏样式</button>
<script>
  var btnTitle = document.querySelector('#J_btn_title');
  var btnImage = document.querySelector('#J_btn_image');
  var btnBgColor = document.querySelector('#J_btn_bgColor');
  var btnBorderColor = document.querySelector('#J_btn_borderColor');
  var btnReset = document.querySelector('#J_btn_reset');

  btnTitle.addEventListener('click', function(){
    ap.setNavigationBar('支付宝-AlipayJSAPI');
  });
  btnImage.addEventListener('click', function(){
    ap.setNavigationBar({
      image: 'https://zos.alipayobjects.com/rmsportal/jFCQxe0XeKTQUfhrQF0o.png'
    });
  });
  btnBgColor.addEventListener('click', function(){
    ap.setNavigationBar({
      title: '设置导航栏背景色',
      backgroundColor: '#ED4A4D'
    });
  });
  btnBorderColor.addEventListener('click', function(){
    ap.setNavigationBar({
      title: '设置导航栏底部边框色',
      borderBottomColor: '#108ee9'
    });
  });
  btnReset.addEventListener('click', function(){
    ap.setNavigationBar({
      title: '重置导航栏样式',
      reset: true
    });
  });
</script>
```

# ap.setOptionButton(OPTION, CALLBACK)

设置导航栏右侧按钮。

## 1. OPTION 参数说明

| 名称             | 类型           | 必填 | 描述  |
|----------------|--------------|----|---|
| items          | Object Array | 否  | 按钮数组，数组中每个项是一个对象。item 的具体配置字段见下表  |
| onClick        | Function     | 否  | 监听 optionButton 点击事件  |
| reset          | Boolean      | 否  | 重置到系统默认按钮，默认为 false。当为 true 时，忽略其他参数                                    |
| preventDefault | Boolean      | 否  | 是否阻止默认的股份功能，指定 iconType 的情况下点击时，会弹分享面板， preventDefault：true 会阻止默认分享面板弹出 |

items 配置项参数说明：

| 名称    | 类型     | 必填 | 描述  |
|-------|--------|----|---|
| title | String | 是  | 按钮标题  |
| color | String | 否  | 按钮标题文字颜色  |
| icon  | String | 否  | 按钮图标，支持 base64  |
| type  | String | 否  | 按钮图标类型，与 title、icon 三选一。支持 user / filter / search / add / settings / scan / info / help / locate / more |
| badge | String | 否  | 按钮红色气泡，默认 -1。其中 0 表示小红点，-1 表示不显示，其他值展示出来  |

onClick 返回值说明：

点击按钮时调用

| 名称    | 类型     | 描述                   |
|-------|--------|----------------------|
| index | Number | 被点击的菜单项的索引，从0开始，从左到右 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn_title" class="btn btn-default">设置文字「完成」 </button>
<button id="J_btn_color" class="btn btn-default">设置文字「颜色」 </button>
<button id="J_btn_type" class="btn btn-default">设置内置类型图标</button>
<button id="J_btn_icon" class="btn btn-default">设置自定义图标</button>
```

<button id="J\_btn\_reset" class="btn btn-default">重置</button>

<script>

```
var btnTitle = document.querySelector('#J_btn_title');
var btnColor = document.querySelector('#J_btn_color');
var btnIcon = document.querySelector('#J_btn_icon');
var btnType = document.querySelector('#J_btn_type');
var btnReset = document.querySelector('#J_btn_reset');
```

```
btnTitle.addEventListener('click', function(){
  ap.setOptionButton({
    items: [{
      title: '完成'
    }],
    onClick: function(data){
      ap.alert('你点击了「完成」，它是第' + data.index + '个 optionButton');
    },
    success: function(res) {
      ap.showToast('设置成功');
    },
    fail: function(res) {
      ap.showToast('设置失败');
    }
  });
});
```

```
btnColor.addEventListener('click', function(){
  ap.setOptionButton({
    items: [{
      title: '颜色',
      color: '#ED4A4D'
    }],
    onClick: function(data){
      ap.alert('你点击了「颜色」，它是第' + data.index + '个 optionButton');
    },
    success: function(res) {
      ap.showToast('设置成功');
    },
    fail: function(res) {
      ap.showToast('设置失败');
    }
  });
});
```

});

```
btnType.addEventListener('click', function(){
  var items = [
    {
      type: 'user'
    },
    {
      type: 'settings'
    }
  ]
});
```

[illegible]

```
        fail: function(res){
            ap.showToast('设置失败');
        }
    });
});
</script>
```

# ap.showNavigationBarLoading()

显示导航栏 loading。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_show" class="btn btn-default">showNavigationBarLoading</button>
<button id="J_btn_hide" class="btn btn-default">hideNavigationBarLoading</button>
<script>
    var btnShow = document.querySelector('#J_btn_show');
    var btnHide = document.querySelector('#J_btn_hide');
    btnShow.addEventListener('click', function(){
        ap.showNavigationBarLoading();
    });
    btnHide.addEventListener('click', function(){
        ap.hideNavigationBarLoading();
    });
</script>
```

# ap.showOptionButton()

显示导航栏右侧按钮。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn_show" class="btn btn-default">显示 </button>
<button id="J_btn_hide" class="btn btn-default">隐藏 </button>
<script>
  var btnShow = document.querySelector('#J_btn_show');
  var btnHide = document.querySelector('#J_btn_hide');

  btnShow.addEventListener('click', function(){
    ap.showOptionButton();
  });
  btnHide.addEventListener('click', function(){
    ap.hideOptionButton();
  });
</script>
```

# ap.showPopupMenu(OPTION | items, CALLBACK)

显示界面右上角弹出的下拉菜单。可直接传入一个数组作为 OPTION.items 参数。

## 1. OPTION 参数说明

| 名称    | 类型           | 必填 | 描述   |
|-------|--------------|----|--|
| items | Object Array | 是  | 菜单数组，数组中每个项是一个对象(也可以是一个字符串，如果是字符串，会当作 item.title 参数)。item 的具体配置字段见下表 |

items 配置项参数说明：

| 名称    | 类型     | 必填 | 描述                                       |
|-------|--------|----|--|
| title | String | 是  | 菜单标题，可直接作为 items 数组元素。                   |
| icon  | String | 是  | 菜单图标，支持 base64                           |
| badge | String | 否  | 菜单红色气泡，默认 -1。其中 0 表示小红点，-1 表示不显示，其他值展示出来 |

## 2. CALLBACK 参数说明

点击弹出菜单项时调用





```
MS4oGEePs0wi9tYy0NlaIs7l3f+r+T/054oNri27ExzLaVejnKDDPv+RajComfBC2yUxZM6b18/iD/bV9CSrvKNT2C+PQH9T7A2AU9T+fYEAK2UiidQfgPR0SxS6fpCQAfqJWkKrxY60BRed
xqUpAmsEPJ44SzZWUAkepQMA0EshIzlvq0sPUobgBcdxb1zWlsjF0mPka9aLIvnu2EuWk9Dkiiri8cSpq1vVCZJGrRHa4FL09LZVimipyH+XwyR2ZgMmc16Dy9GGuS5kUZ4Lm42S7g1umNDH
l+0uLjIXbhWf/Ry/LIRZ9q5g3Brb38+vUS7pmGK9lTJlSfdY7DE0CF+Bz/KPZm0o5o31Bh8xZj0QYwR6+0zoanG9VXfrIFBKKnG99aepIMhb6pzUPDMNnIXeqrzl8pvm4BbT0BykpJn19PLz
Ha1moZ0TrJP4XJ2jFCViJN/oZdwmjiJqX5x/FtBh99MPC+YHh2nKP0VcNPUhHAqUQcZNoeidpZAU4/En1vpJfXHe/vIWmtL3bxonH2ZTCpgKZpMQY0jqRw0LTTRAh1WmmaFLokmSQN9qenvf
/7fKNXVs3zjRk70iZUetQq4ynl6jHfQxPYkxvPjgaBuWDBwH1oaZBHG0Ds0vINxapV0i1wVPIQPMvn+yrg75SKZDDx0LZq66KhgFoyrJoZdIwqYTeESo/6gv/RHigsqu04PrWwzz9R0C40AA
AAASUV0RK5CYII=',
      badge: '0'
    }
  ]
}, function(res){
  ap.showToast('你点击了第 ' + res.index + ' 个菜单');
});
});
</script>
```

## 4. 其他说明

- 如果设置了 icon 那么每一个菜单项都要求有 icon，否则都没有 icon。

# ap.alert(OPTION | content, CALLBACK)

alert 警告框。可直接传入一个字符串作为 OPTION.content 参数。

## 1. OPTION 参数说明

| 名称         | 类型     | 必填 | 描述          |
|------------|--------|----|-------------|
| title      | String | 否  | alert框的标题   |
| content    | String | 否  | alert框的内容   |
| buttonText | String | 否  | 按钮文字，默认‘确定’ |

## 2. CALLBACK 参数说明

点击按钮后调用

## 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">alert</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.alert({
      title: '亲',
      content: '您有新的快递消息',
      buttonText: '我知道了'
    }, function(){
      // 可直接传入一个字符串当作 OPTION.content 参数
      ap.alert('用户点击了「我知道了」');
    });
  });
</script>
```

#### 4. 其他说明

- ap.alert 不像window.alert，会不阻塞 JS 代码执行，即先后弹2个的话，最后看到的是后弹的那个。

## ap.confirm(OPTION | content, CALLBACK)

confirm 确认框。可直接传入一个字符串作为 OPTION.content 参数。

#### 1. OPTION 参数说明

| 名称                | 类型     | 必填 | 描述            |
|-------------------|--------|----|---------------|
| title             | String | 否  | confirm 框的标题  |
| content           | String | 否  | confirm 框的内容  |
| confirmButtonText | String | 否  | 确定按钮文字，默认’确定’ |
| cancelButtonText  | String | 否  | 取消按钮文字，默认’取消’ |

#### 2. CALLBACK 参数说明

点击相应按钮后调用

| 名称      | 类型      | 描述  |
|---------|---------|---|
| confirm | Boolean | 用户选择结果。点击 confirmButton 为 true，点击cancelButton 为 false |

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">confirm</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.confirm({
      title: '温馨提示',
      content: '您是否想查询快递单号：\n1234567890',
      confirmButtonText: '马上查询',
      cancelButtonText: '暂不需要'
    }, function(result){
      ap.alert(result.confirm);
    });
  });
</script>
```

### 4. 其他说明

- ap.confirm 和 ap.alert 一样，它也不是阻塞式的，即先后弹2个的话，最后看到的是后弹的那个。

## ap.hideLoading()

隐藏加载提示。

### 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">showLoading</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.showLoading({
      content: '2秒后消失',
      delay: 1000
    });
    setTimeout(function(){
      ap.hideLoading();
    }, 3000);
  });
</script>
```

## 2. 其他说明

- 显示 loading 后，会覆盖整个h5页面，页面元素不能交互。

# ap.hideToast()

隐藏弱提示。

## 1. 代码示例

<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2>点击以下按钮查看不同效果</h2>

<button id="J\_showToast" class="btn btn-default">显示 toast</button>  
<button id="J\_hideToast" class="btn btn-default">隐藏 toast</button>

```
<script>
  var showBtn = document.querySelector('#J_showToast');
  var hideBtn = document.querySelector('#J_hideToast');
  showBtn.addEventListener('click', function(){
    ap.showToast('显示中...');
  });
  hideBtn.addEventListener('click', function(){
    ap.hideToast();
  });
</script>
```

# ap.showActionSheet(OPTION, CALLBACK)

显示操作菜单。

## 1. OPTION 参数说明

| 名称               | 类型           | 必填 | 描述             |
|------------------|--------------|----|----------------|
| title            | String       | 否  | 菜单标题           |
| items            | String Array | 是  | 菜单按钮的文字数组      |
| cancelButtonText | String       | 否  | 取消按钮文案，默认为「取消」 |

## 2. CALLBACK 参数说明

点击菜单按钮时调用

| 名称    | 类型     | 描述                           |
|-------|--------|------------------------------|
| index | Number | 被点击的按钮的索引，从0开始。点击取消或蒙层时返回 -1 |

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">显示 ActionSheet</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.showActionSheet({
      title: '支付宝-ActionSheet',
      items: ['菜单一', '菜单二', '菜单三']
    }, function(res){
      var btn = res.index === -1 ? '取消' : '第' + res.index + '个';
      ap.showToast('你点了' + btn + '按钮' );
    });
  });
</script>
```

# ap.showLoading(OPTION | content)

显示加载提示。可直接传入一个字符串作为 OPTION.content 参数。

### 1. OPTION 参数说明

| 名称      | 类型     | 必填 | 描述   |
|---------|--------|----|--|
| content | String | 否  | loading 的文字提示                                    |
| delay   | Number | 否  | 延迟显示，单位 ms，默认 0。如果在此时间之前调用了 ap.hideLoading 则不会显示 |

### 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">showLoading</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.showLoading({
      content: '2秒后消失',
      delay: 1000
    });
    setTimeout(function(){
      ap.hideLoading();
    }, 3000);
  });
</script>
```

### 3. 其他说明

- 显示 loading 后，会覆盖整个h5页面，页面元素不能交互。

# ap.showToast(OPTION | content, CALLBACK)

显示一个弱提示，可选择多少秒之后消失。可直接传入一个字符串作为 OPTION.content 参数。

### 1. OPTION 参数说明

| 名称       | 类型     | 必填 | 描述   |
|----------|--------|----|--|
| content  | String | 是  | 文字内容   |
| type     | String | 否  | toast 类型，展示相应图标，默认 none，支持 success / fail / exception / none’。其中 exception 类型必须传文字信息 |
| duration | Number | 否  | 显示时长，单位为 ms，默认 2000  |

### 2. CALLBACK 参数说明

toast消失后调用

### 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2>点击以下按钮查看不同效果</h2>

<button class="toast-btn btn btn-default">success</button>
<button class="toast-btn btn btn-default">fail</button>
<button class="toast-btn btn btn-default">exception</button>
<button class="toast-btn btn btn-default">none</button>
<button id="J_btn" class="btn btn-default">展示3秒并执行回调</button>

<script>
  var btns = document.querySelectorAll('.toast-btn');
  var btn = document.querySelector('#J_btn');
  var contents = {
    success: '操作成功',
    fail: '系统繁忙',
    exception: '网络异常',
    none: '忙不过来了, 客官请稍候'
  };
  [].forEach.call(btns, function(btn){
    btn.addEventListener('click', function(){
      var type = this.innerHTML;
      ap.showToast({
        content: contents[type],
        type: type
      });
    });
  });
  btn.addEventListener('click', function(){
    ap.showToast({
      content: '请稍候...',
      duration: 3000
    }, function(){
      ap.alert('toast消失了');
    });
  });
</script>
```

### 4. 其他说明

- 安卓需要注意，如果系统通知关闭，则此弹框不会出现。



# ap.allowPullDownRefresh(OPTION | allow)

下拉刷新开关。可直接传入一个布尔值作为 OPTION.allow 参数。

## 1. OPTION 参数说明

| 名称    | 类型      | 必填 | 描述               |
|-------|---------|----|------------------|
| allow | Boolean | 否  | 是否允许下拉刷新，默认 true |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2 class="am-ft-center">下拉试试</h2>
<script>
  ap.allowPullDownRefresh(false);
  ap.onPullDownRefresh(function(res){
    if(!res.refreshAvailable){
      ap.alert({
        content: '刷新已禁止',
        buttonText: '恢复'
      }, function(){
        ap.allowPullDownRefresh(true);
        ap.showToast('刷新已恢复')
      });
    }
  });
</script>
```

# ap.offPullDownRefresh(CALLBACK)

停止监听页面下拉刷新事件。

## 1. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<h2 class="am-ft-center">下拉试试</h2>
```

```
<script>
  var handler = function(res){
    if(!res.refreshAvailable){
      ap.alert('刷新已禁止');
    }
    ap.offPullDownRefresh(handler);
  };
  ap.allowPullDownRefresh(false);
  ap.offPullDownRefresh(handler);
</script>
```

# ap.onPullDownRefresh(CALLBACK)

当页面下拉准备刷新时触发。

## 1. CALLBACK 参数说明

| 名称               | 类型      | 描述   |
|------------------|---------|--|
| refreshAvailable | Boolean | 是否可刷新。可通过ap.allowPullDownRefresh() (../ui/allowPullDownRefresh.html)设置此返回值 |

## 2. 代码示例

<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<h2 class="am-ft-center">下拉试试</h2>

```
<script>
  ap.allowPullDownRefresh(false);
  ap.onPullDownRefresh(function(res){
    if(!res.refreshAvailable){
      ap.alert({
        content: '刷新已禁止',
        buttonText: '恢复'
      }, function(){
        ap.allowPullDownRefresh(true);
        ap.showToast('刷新已恢复')
      });
    }
  });
</script>
```

# ap.share(OPTION, CALLBACK)

分享内容。

## 1. OPTION 参数说明

| 名称            | 类型      | 必填 | 描述                       |
|---------------|---------|----|--------------------------|
| title         | String  | 是  | 分享标题                     |
| content       | String  | 是  | 分享内容                     |
| url           | String  | 是  | 分享链接                     |
| image         | String  | 否  | 分享图片链接                   |
| captureScreen | Boolean | 否  | 是否截屏，默认 false            |
| showToolBar   | Boolean | 否  | 是否显示分享面板第二行的工具按钮，默认 true |

## 2. CALLBACK 参数说明

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

| 名称          | 类型      | 描述    |
|-------------|---------|-------|
| shareResult | Boolean | 分享的结果 |
| channelName | String  | 分享渠道  |

### 3. 错误码描述

| error | 描述      |
|-------|---------|
| 10    | 分享失败或取消 |

### 4. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">打开分享面板</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.share(function(result){
      ap.alert(result.shareResult);
    });
  });
</script>
```

# ap.getAuthCode(OPTION | scopes, CALLBACK)

获取授权码，具体请参考用户授权接入 (../../fe/auth.html)。可直接传入一个字符串作为 OPTION.scopes 参数的第一个元素。

### 1. OPTION 参数说明

| 名称     | 类型           | 必填 | 描述   |
|--------|--------------|----|--|
| scopes | String/Array | 否  | 授权类型，默认 auth_base。支持 auth_base(静默授权)/auth_user(主动授权) |

## 2. CALLBACK 参数说明

| 名称       | 类型     | 描述  |
|----------|--------|-----|
| authCode | String | 授权码 |

## 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">getAuthCode</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.getAuthCode('auth_user', function(res){
      ap.alert(res.authCode);
    });
  });
</script>
```

# ap.getAuthUserInfo(CALLBACK)

获取授权用户信息，具体请参考用户授权接入 (../../fe/auth.html)。

## 1. CALLBACK 参数说明

| 名称       | 类型     | 描述     |
|----------|--------|--------|
| nickName | String | 用户昵称   |
| avatar   | String | 用户头像链接 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">getAuthUserInfo</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.getAuthCode('auth_user', function(res){
      ap.getAuthUserInfo(function(res){
        ap.alert(res.nickName);
      });
    });
  });
</script>
```

# ap.getServerTime(CALLBACK)

获取当前服务器时间的毫秒数

## 1. CALLBACK 参数说明

| 名称   | 类型     | 描述        |
|------|--------|-----------|
| time | Number | 服务器时间的毫秒数 |

## 2. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>

<button id="J_btn" class="btn btn-default">获取服务器时间</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.getServerTime(function(res){
      ap.alert(res.time)
    });
  });
</script>
```

# ap.startBizService(OPTION | name, CALLBACK)

启动某项支付宝业务服务。可直接传入一个字符串作为 OPTION.name 参数。

## 1. OPTION 参数说明

| 名称     | 类型     | 必填 | 描述     |
|--------|--------|----|--------|
| name   | String | 是  | 业务服务名称 |
| params | Object | 否  | 业务服务参数 |

## 2. CALLBACK 参数说明

其他字段业务服务自行定义

| 名称           | 类型     | 描述         |
|--------------|--------|------------|
| resultStatus | String | 结果状态码，详见下表 |

### resultStatus 状态码

| resultStatus | 描述        |
|--------------|-----------|
| 6001         | 用户取消了业务流程 |
| 6002         | 网络异常      |

## 3. 代码示例

```
<script src="https://a.alipayobjects.com/g/h5-lib/alipayjsapi/3.0.2/alipayjsapi.inc.min.js"></script>
```

```
<button id="J_btn" class="btn btn-default">启动芝麻信用借还服务</button>
<script>
  var btn = document.querySelector('#J_btn');
  btn.addEventListener('click', function(){
    ap.startBizService('zm-borrow', function(res){
      ap.alert(res.resultStatus);
    });
  });
</script>
```

## 4. 其他说明

- 此接口支付宝 10.0.15 及以上支持

## 1. 常用资源内置

这些常用组件已经包含在容器中，使用 `<script src="{{src}}"></script>` 不会产生网络流量。

如需要其他资源，可以自己打包内置在应用里。

- AntUI 10.0.0版本: <https://a.alipayobjects.com/g/antui/antui/10.0.0/rem/antui.css>
- zepto: <https://a.alipayobjects.com/amui/zepto/1.1.3/zepto.js>
- fastclick: <https://a.alipayobjects.com/static/fastclick/1.0.6/fastclick.min.js>
- antbridge.js 1.1.4版本: <https://a.alipayobjects.com/g/component/antbridge/1.1.4/antbridge.min.js>
- Anima-yocto 1.1.10版本: <https://a.alipayobjects.com/anima-yocto/1.1.10/index.js>
- swiper 3.2.7版本: <https://as.alipayobjects.com/g/component/swiper/3.2.7/swiper.min.js> ,  
<https://as.alipayobjects.com/g/component/swiper/3.2.7/swiper.min.css>
- react: <https://as.alipayobjects.com/g/component/react/15.3.2/react.min.js> , <https://as.alipayobjects.com/g/component/react/15.3.2/react-dom.min.js>
- vuejs 1.0.26: <https://as.alipayobjects.com/g/h5-lib/vue/1.0.26/vue.min.js>
- vuejs 2.1.6: <https://a.alipayobjects.com/g/h5-lib/vue/2.1.6/vue.min.js>
- seedjs 3.1.0: <https://a.alipayobjects.com/g/animajs/mtracker/3.1.0/seed.js>
- es6: <https://as.alipayobjects.com/g/component/es6-promise/3.2.2/es6-promise.min.js> ,



`https://as.alipayobjects.com/g/component/fetch/1.0.0/fetch.min.js`

通用错误图片：

- 加载出错：<https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/error.png> (<https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/error.png>)
- 系统繁忙：[https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/system\\_busy.png](https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/system_busy.png) ([https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/system\\_busy.png](https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/system_busy.png))
- 网络错误：[https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/no\\_network.png](https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/no_network.png) ([https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/no\\_network.png](https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/no_network.png))
- 空白页面：[https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/blank\\_page.png](https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/blank_page.png) ([https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/blank\\_page.png](https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/blank_page.png))
- 404页面：<https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/404.png> (<https://a.alipayobjects.com/g/antui/antui-img/1.0.0/page-result/404.png>)

## 其他参考资料

### 1. 开发库

- Ant Design Mobile （React 推荐首选） (<https://mobile.ant.design/>)
- AntUI (<http://am-team.github.io/antui/nav.html>)
- 支付宝开放平台文档中心 (<https://doc.open.alipay.com/>)

### 2. Javascript

- Javascript Garden (<http://bonsaiden.github.io/JavaScript-Garden/>)
- The Basics Of ES6 Generators (<http://davidwalsh.name/es6-generators>)
- Understanding ECMAScript 6 (<https://leanpub.com/understandingses6/read>)
- Javascript online free books (<http://jsbooks.revolunet.com/>)
- 阮一峰ES6之书 (<http://es6.ruanyifeng.com/>)

Since 8.2

# exitApp

exitApp用于退出当前应用所有的页面

## 1. 使用方法

```
AlipayJSBridge.call('exitApp');
```

## 2. 代码演示

### 2.1. 退出当前的页面

```
<h1>点击退出当前页面</h1>
<a href="#" class="btn btn-default J_demo">执行</a>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('a').addEventListener('click', function(){
    AlipayJSBridge.call('exitApp');
  });
});
</script>
```

### 2.2. 页面跳转完整实例

```
<h1>请点击下面按钮来进行页面间跳转</h1>
<h3></h3>
<a href="javascript:void(0)" class="new">新开当前页面</a>
<a href="javascript:void(0)" class="back">返回一级</a>
<a href="javascript:void(0)" class="popTo">通过popTo退2级</a>
```

<a href="javascript:void(0)" class="exit">关闭所有页面</a>

```
<script>
function getQuery() {
    let match;
    const urlParams = {};
    const pl = /\+/g;  // Regex for replacing addition symbol with a space
    const search = /(?:[&=]+)=?(?:[^\&]*)/g;
    const decode = function (s) {
        return decodeURIComponent(s.replace(pl, ' '));
    };
    const query = decodeURI(window.location.search.substring(1));

    while (match = search.exec(query)) {
        urlParams[decode(match[1])] = decode(match[2]);
    }

    return urlParams;
};

var query = getQuery();
var depth = (+query.depth) || 0;

document.querySelector('h3').innerHTML = '当前页面深度: ' + depth;

function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('.new').addEventListener('click', function(){
        AlipayJSBridge.call('pushWindow', {
            url: location.pathname + '?depth=' + (1+depth),
        });
    });

    document.querySelector('.back').addEventListener('click', function(){
        AlipayJSBridge.call('popWindow', {
            data: {
                method: 'popWindow',
                depth: depth,
            }
        });
    });
});
```

```
document.querySelector('.popTo').addEventListener('click', function(){
    AlipayJSBridge.call('popTo', {
        index: -2,
        data: {
            method: 'popTo',
            depth: depth,
        }
    }, function(e){
        if(e.error) {
            alert('发生错误: ' + JSON.stringify(e));
        }
    });
});

document.querySelector('.exit').addEventListener('click', function(){
    AlipayJSBridge.call('exitApp');
});

document.addEventListener('resume', function (event) {
    alert('页面回退时带过来的内容: ' + JSON.stringify(event.data));
});
</script>
```

### 3. 使用注意

- 对于没有申请过appId的页面，都是以20000067这个appId运行，因此，任何页面内调用 exitApp ，所有页面都会关闭。

Since 8.1

# popTo

popTo可以一次回退多级页面

### 1. 使用方法

```
// 关闭当前打开的页面
AlipayJSBridge.call('popTo', {
    index: -1
});
```

## 2. 代码演示

### 2.1. 关闭当前页面并且传递数据

```
<h1>第二步，点击执行关闭当前页面并返回数据</h1>
<a href="#" class="btn btn-default J_demo">执行</a>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('a').addEventListener('click', function(){
    // 传递的data对象将会被即将露出的页面通过resume事件接收
    AlipayJSBridge.call('popTo', {
      index: -1, // 回退到上一个页面，假如这个时候没有上一个页面，就会报错
      data: { // 特别强调：data是字典，不是数组
        from: location.href,
        info: 'xxxx'
      }
    }, function(e) { // 添加回调，因为popTo不一定会成功（当前页面是唯一打开的页面的时候，会报错）
      alert(JSON.stringify(e));
    });
  });
});
</script>
```

### 2.2. 通过url返回到指定的页面

```
<h1>返回到指定的url</h1>
<h3></h3>
<a href="javascript:void(0)" class="btn btn-default J_new_window">新窗口打开当前页面</a>
<a href="javascript:void(0)" class="btn btn-default J_demo">返回2级</a>
<script>
var query = getQuery();
var depth = (+query.depth) || 0;

document.querySelector('h3').innerHTML = '当前页面深度： ' + depth;

function ready(callback) {
```

```

// 如果jsbridge已经注入则直接调用
if (window.AlipayJSBridge) {
    callback && callback();
} else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
}
}
ready(function(){
    document.querySelector('.J_demo').addEventListener('click', function(){
        // 返回2级
        var queryString = depth -2 > 0 ? '?depth=' + (depth -2) : '';
        AlipayJSBridge.call('popTo', {
            url: location.pathname + queryString,
        }, function(e){
            alert(JSON.stringify(e));
        });
    });

    document.querySelector('.J_new_window').addEventListener('click', function(){
        AlipayJSBridge.call('pushWindow', {
            url: location.pathname + '?depth=' + (1+depth),
        });
    });
});

// 返回当前queryString的对象: {query: 1}
function getQuery() {
    var match;
    var urlParams = {};
    var pl = /\+/g; // Regex for replacing addition symbol with a space
    var search = /([^\&=]+)=?([^\&]*)/g;
    var decode = function (s) {
        return decodeURIComponent(s.replace(pl, ' '));
    };
    var query = decodeURI(window.location.search.substring(1));

    while (match = search.exec(query)) {
        urlParams[decode(match[1])] = decode(match[2]);
    }

    return urlParams;
};
</script>

```

## 2.3. 通过urlPattern返回到符合正则匹配的页面

<h1>返回到符合某个规则的url</h1>

```
<h3></h3>
<a href="javascript:void(0)" class="btn btn-default J_new_window">新窗口打开当前页面</a>
<a href="javascript:void(0)" class="btn btn-default J_demo">返回</a>
<script>
var query = getQuery();
var depth = (+query.depth) || 0;

document.querySelector('h3').innerHTML = '当前页面深度: ' + depth;

function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('.J_demo').addEventListener('click', function(){
        AlipayJSBridge.call('popTo', {
            urlPattern: 'pop-to-url-pattern.html',
        }, function(e){
            alert(JSON.stringify(e));
        });
    });

    document.querySelector('.J_new_window').addEventListener('click', function(){
        AlipayJSBridge.call('pushWindow', {
            url: location.pathname + '?depth=' + (1+depth),
        });
    });
});

// 返回当前queryString的对象: {query: 1}
function getQuery() {
    var match;
    var urlParams = {};
    var pl = /\+/g; // Regex for replacing addition symbol with a space
    var search = /([^\&=]+)=?([^\&]*)/g;
    var decode = function (s) {
        return decodeURIComponent(s.replace(pl, ' '));
    };
    var query = decodeURI(window.location.search.substring(1));

    while (match = search.exec(query)) {
        urlParams[decode(match[1])] = decode(match[2]);
    }

    return urlParams;
}
```

```
};
</script>
```

### 3. API

```
AlipayJSBridge.call('popTo',{
  index, url, urlPattern
}, fn)
```

index, url, urlPattern 这是3种查询模式， 不应该被同时使用。

#### 3.1. 入参

| 名称         | 类型       | 描述                                       | 必选 | 默认值 | 版本 |
|------------|----------|--|----|-----|----|
| index      | int      | 目标界面在会话界面栈中的索引；如果小于零，则将与当前界面的index相加     | Y  |     |    |
| url        | string   | 目标界面的完整URL                               | Y  |     |    |
| urlPattern | string   | 目标界面的URL匹配表达式（URL如果包含urlPattern， 匹配成功） Y |    |     |    |
| fn         | function | 操作成功时，回调可能不被调用；操作失败时，回调函数执行              | N  |     |    |

#### 3.2. 出参

| 名称     | 类型        | 描述                         |
|--------|-----------|----------------------------|
| result | undefined | 操作成功时，回调可能不被调用；result不应被使用 |

#### 3.3. 错误

| error | 描述   |
|-------|--|
| 10    | 未配置参数；<br>无效的index；<br>未匹配url；<br>未匹配urlPattern； |

### 4. 使用注意



- 一般情况下， popTo 用于分多步才能完成的场景，比如用户信息完善时回退，三级地址选择后返回等。
- 如果通过url或者urlPattern来返回， popTo 会返回到离当前页面最远的页面，也就是栈底。同时不会去检测当前页面的url是否符合。
- popTo 时所带的data如何被接收？ 请查看resume事件 (../events/resume.html)

# popWindow

popWindow用来关闭当前页面

## 1. 使用方法

```
// 关闭当前打开的页面
AlipayJSBridge.call('popWindow');
```

## 2. 代码演示

### 2.1. 关闭当前页面

```
<h1>关闭当前页面</h1>
<a href="#" class="btn btn-default J_demo">执行</a>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('a').addEventListener('click', function(){
    AlipayJSBridge.call('popWindow');
  });
});
</script>
```

### 2.2. 关闭当前页面并且传递数据

```
<h1>关闭当前页面并返回数据</h1>
<a href="#" class="btn btn-default J_demo">执行</a>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('a').addEventListener('click', function(){
    // 传递的data对象将会被即将露出的页面通过resume事件接收
    AlipayJSBridge.call('popWindow', {
      data: {
        from: location.href,
        info: 'xxxx'
      }
    });
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('popWindow',{
  data
})
```

#### 3.1. 入参

| 名称   | 类型     | 描述           | 必选 | 默认值 | 版本 |
|------|--------|--------------|----|-----|----|
| data | object | 传递给即将露出页面的内容 | N  |     |    |

### 4. 使用注意

- closeWebview 是 popWindow 的别名，使用方式是 AlipayJSBridge.call('closeWebview')
- popWindow 时所带的data如何被接收？ 请查看resume事件 (../jsapi/events/resume.html)

# pushWindow

pushWindow用来打开一个新的页面，自带转场动画。

与location.href的区别，类同于PC浏览器的新开标签页，每个window都是一个新的标签页，因此原页面仅仅是被压到后台，状态始终保持，JS也会继续运行。

## 1. 使用方法

```
// 打开淘宝首页，自动读取title，并且去除右边菜单
AlipayJSBridge.call('pushWindow', {
  url: 'https://m.taobao.com/',
  param: {
    readTitle: true,
    showOptionsMenu: false
  }
});
```

## 2. 代码演示

### 2.1. 打开淘宝首页，去除右边菜单

```
<h1>打开淘宝首页</h1>
<a class="btn btn-default J_demo">执行</a>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('a').addEventListener('click', function(){
    // 打开淘宝首页, 自动读取title, 并且去除右边菜单
    AlipayJSBridge.call('pushWindow', {
      url: 'https://m.taobao.com/',
      param: {
        readTitle: true,
        showOptionsMenu: false
      }
    });
  });
});
</script>
```

## 2.2. 用schema的方式打开页面

9.9.5开始支持，此前使用该方式会导致先打开一个空白页面。

```
<h1>打开指定app</h1>
<a class="btn btn-default J_demo">执行</a>
<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('a').addEventListener('click', function(){
        AlipayJSBridge.call('pushWindow', {
            url: 'alipays://platformapi/startapp?appId=20000981',
        });
    });
});
</script>
```

## 2.3. 打开时设置透明标题栏

```
<h1>打开淘宝首页</h1>
<a class="btn btn-default J_demo">执行</a>
<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('a').addEventListener('click', function(){
        AlipayJSBridge.call('pushWindow', {
            url: 'https://m.taobao.com',
            param: {
                transparentTitle: 'auto'
            }
        });
    });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('pushWindow', {
  url, param
})
```

#### 3.1. 入参

| 名称    | 类型         | 描述                 | 必选 | 默认值 | 版本 |
|-------|------------|--------------------|----|-----|----|
| url   | string     | 要打开的url            | Y  |     |    |
| param | dictionary | 支持的key/value对下面的表格 | N  | {}  |    |

#### 3.2. param参数详解

| 名称                   | 类型     | 描述   | 默认值              | 版本                                  |
|----------------------|--------|--|------------------|-------------------------------------|
| defaultTitle         | string | 默认标题，在页面第一次加载之前显示在标题栏上   | ''               |                                     |
| showLoading          | bool   | 是否在页面加载前显示全局菊花   | false            |                                     |
| readTitle            | bool   | 是否读取网页标题显示在titleBar上   | true             |                                     |
| pullRefresh          | bool   | 是否支持下拉刷新<br>只有集团域/本地文件允许设置为true  | false            | since 8.2                           |
| toolbarMenu          | string | JSON字符串，更多的菜单项列表（放在分享、字号、复制链接后面）<br>例：{“menus”:[{“name”:”恭喜”,“icon”:”H5Service.bundle/h5_popovermenu_share”,“action”:”hello”},{“name”:”发财”,“icon”:”H5Service.bundle/h5_popovermenu_abuse”,“action”:”world”}]]} | ''               | since 8.2                           |
| canPullDown          | bool   | 页面是否支持下拉（显示出黑色背景或者域名）<br>只有.alipay.com/.alipay.net/本地文件允许设置为false<br>9.9.5废弃， 使用’allowsBounceVertical’替代   | true             | android since 8.3<br>ios since 8.4, |
| allowsBounceVertical | bool   | 页面是否支持纵向拽拉超出实际内容，android只支持下拉（显示出背景或者域名）<br>只有.alipay.com/.alipay.net/本地文件允许设置为false   | true             | since 9.9.3                         |
| bounceTopColor       | int    | 下拉超出时，顶部间缝颜色（十进制，例如：bc=16775138）   | 不设置时与9.9.3之前显示一致 | since 9.9.3                         |
| bounceBottomColor    | int    | 上拉超出时，底部间缝颜色（十进制，例如：bc=16775138）   | 不设置时与9.9.3之前显示一致 | since 9.9.3(only IOS)               |
| showTitleLoading     | bool   | 是否在TitleBar的标题左边显示小菊花  | false            | since 8.6                           |

|                    |        |  |         |                           |
|--------------------|--------|--|---------|---------------------------|
| preRpc             | string | 在页面启动参数里配置rpc请求的参数preRpc（需要使用utf-8编码），在打开页面的同时请求rpc。详细使用参考RPC接口介绍  |         | since 9.3                 |
| delayRender        | bool   | 是否启动延迟渲染功能<br>注：本功能目前由线上开关控制，若线上开关打开，且指定启动参数为YES或TRUE则生效   | false   | since 9.3.1(only Android) |
| transparentTitle   | bool   | (不能与titleBarColor同时使用)<br>always/auto 如果transparentTitle为字符串always，则当前页面上下滚动时，titlebar一直全透明；<br>当transparentTitle值为auto，当页面往下滚动时，透明度不断增加，直到scrollTop等于80pt时变成完全不透明，此时页面再往上滚动则反之，透明度不断减小直到回到顶部时变成完全不透明。<br>如果个页面不需要透明效果，则需要用pushWindow的param参数重新指定transparentTitle为”none” | ‘’      | since 9.5.1               |
| titleBarColor      | int    | (9.9版本以下不能与transparentTitle同时使用)自定义titlebar的背景色，（十进制，例如：bc=16775138）   |         |                           |
| scrollDistance     | int    | 在 transparentTitle auto的情况下，滑动到透明度为0.96的距离   | iOS(80) | since 9.9                 |
| startMultApp       | bool   | 是否启动相同appID的应用   | false   | since 9.9                 |
| titleImage         | string | 所要使用的title图片地址，请上传一张3X PNG 图，只影响当前的VC，pushWindow不会自动传递此参数，为了更好的体验可以把图放在全局运营资源包中  | ‘’      | since 9.9.5               |
| closeCurrentWindow | bool   | 打开窗口的同时，关闭当前window。 <b>警告：iOS上发现有缺陷:被打开的新页面会导致无法再通过exitApp或startApp+closeCurrentApp来关闭，这种使用场景请勿使用pushWindow+closeCurrentWindow</b>   | false   | since 9.9.2               |

### 3.3. pushWindow参数默认继承

| 名称              | 继承 | 备注            |
|-----------------|----|---------------|
| url             | Y  |               |
| defaultTitle    | Y  |               |
| backBehavior    | Y  | 优先用户指定, 否则pop |
| showLoading     | Y  |               |
| readTitle       | Y  |               |
| pullRefresh     | Y  |               |
| toolbarMenu     | Y  |               |
| showProgress    | Y  |               |
| defaultSubtitle | Y  |               |
| backgroundColor | Y  |               |
| canPullDown     | Y  |               |

|                  |   |  |
|------------------|---|--|
| showOptionsMenu  | Y |  |
| showTitleLoading | Y |  |
| showDomain       | Y |  |

## 4. 如何正确理解pushWindow与location.href

许多新手都会不太理解pushWindow和location.href的区别，同时也会疑惑为什么resume没有被触发之类的问题。 本篇话题只需要几分钟就能把这几个功能解释清楚。

- 首先把Nebula容器理解为一个PC浏览器。
- 「Window」 可以理解为「标签页」； location.href就是正常的a标签跳转。
- pushWindow实际上就是新开了一个「标签页」， 并且把之前那一页压到下面， push出来的这页放在顶层展现。此时被压到下面的上一页所有状态均保留。
- location.href就是依然停留在这个「标签页」， 直接做页面跳转
- 当pushWindow出来的「标签页」被关闭(pop)时， 如果之前还有window存在， 那么之前那个window就会恢复显示， 并触发resume

## 5. 使用注意

- pushWindow 打开url页面的时候不会关闭已经存在的页面， 注意打开个数， 不要开太多影响性能。**建议同一个应用 pushWindow 层级不要超过5层， 否则会影响用户体验以及有可能导致应用crash。**
- pushWindow 打开schema页面的时候会关闭这个appId已经存在的所有页面， 然后打开一个新的。 比如已经打开了200000981， 然后再使用 pushWindow 打开这个appId， 那么当前的20000981页面会被关闭
- 使用schema的时候， 假如参数不生效， 请不要担心， 因为每个app有配置会覆盖传入的参数。 比如上述的20000981， 传入 transparentTitle: 'auto' 是没有效果的。
- 9.9.2版本开始， 可以使用 closeCurrentWindow 参数来替代 window.location.href 跳转,而且有动画效果支持， 这样可以实现关闭当前window的同时打开一个新页面。但在iOS发现有缺陷， 被打开的新页面会导致无法再通过exitApp或startApp+closeCurrentApp来关闭自身， 这种使用场景请勿使用 pushWindow+closeCurrentWindow。

Since 8.0

# AlipayJSBridgeReady



window.onload以后，容器会初始化，产生一个全局变量AlipayJSBridge, 然后触发此事件

AlipayJSBridge注入是一个异步过程，因此尽可能先监听该事件后再call接口。

强烈建议换用antBridge.js来简化JSAPI调用。

## 1. 使用方法

```
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
```

## 2. 代码演示

### 2.1. bridge入口标准写法

```
<h1>bridge使用方法</h1>

<script>
function ready(callback) {
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}

ready(function(){
  alert('bridge ready');
});
</script>
```

Since 9.6

# h5NetworkChange

网络环境发生变化时，会触发这个事件，如需详细网络环境信息，请使用 `getNetworkType` 来获取

## 1. 使用方法

```
document.addEventListener('h5NetworkChange', function (e) {  
    alert("网络环境发生变化，可调用getNetworkType接口获取详细信息");  
}, false);
```

## 2. 代码演示

### 2.1. 基本功能

<h1>请切换手机网络模式，比如wifi和4g</h1>

```
<script>  
document.addEventListener('h5NetworkChange', function (e) {  
    // **注意安卓下addEventListener里的e不能用JSON.stringify，这里的e在UC下不是一个有效的可序列化对象**  
  
    alert("我收到网络变化了");  
    // 网络环境发生变化，可调用getNetworkType接口获取详细信息  
    AlipayJSBridge.call('getNetworkType', function (result) {  
        alert(JSON.stringify(result));  
    });  
}, false);  
</script>
```

## 3. 使用注意

- 3G切换到4G时系统不会发出网络切换通知。 4G切到3G时，会先从4G跳到2G，然后再从2G跳3G，因此会通知两次
- 安卓下回调函数内部不能对e进行JSON.stringify操作，这个e对象不是可被序列化的。

Since 8.0

# optionMenu

当调用setOptionsMenu接口，自定义了导航栏右上角按钮以后，点击按钮时触发该事件

## 1. 使用方法

```
document.addEventListener('optionMenu', function (e) {  
    alert("option menu clicked");  
}, false);
```

## 2. 代码演示

### 2.1. 基本功能演示

```
<h1>请点击右上角菜单查看效果</h1>  
<script>  
function ready(callback) {  
    // 如果jsbridge已经注入则直接调用  
    if (window.AlipayJSBridge) {  
        callback && callback();  
    } else {  
        // 如果没有注入则监听注入的事件  
        document.addEventListener('AlipayJSBridgeReady', callback, false);  
    }  
}  
ready(function(){  
    AlipayJSBridge.call('setOptionsMenu', {  
        title : '按钮',  
        redDot : '5', // -1表示不显示, 0表示显示红点, 1-99表示在红点上显示的数字  
        color : '#ff00ff00', // 必须以#开始ARGB颜色值  
    });  
});  
  
document.addEventListener('optionMenu', function (e) {  
    alert("optionMenu clicked!");  
}, false);  
</script>
```

2.2. 多个optionMenu情况的事件

```
<h1>请点击右上角每个菜单查看效果</h1>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  AlipayJSBridge.call('setOptionsMenu', {
    // 显示的时候是从后往前显示的
    menus: [{
      icontype: 'scan',
      redDot: '-1', // -1表示不显示, 0表示显示红点, 1-99表示在红点上显示的数字
    }, {
      icontype: 'user',
      redDot: '-1', // -1表示不显示, 0表示显示红点, 1-99表示在红点上显示的数字
    }],
    override: true //在需要设置多个option的情况下, 是否保留默认的optionMenu
  });

  // 必须强制调用一把刷新界面
  AlipayJSBridge.call('showOptionsMenu');
});

document.addEventListener('optionMenu', function (e) {
  alert(JSON.stringify(e));
}, false);
</script>
```

3. 使用注意

- 假如没有设置过optionMenu，那么这个事件是无法被触发的，也就是说点击默认的...是无法触发optionMenu事件的。

当一个webview界面不可见时(被压入后台、锁屏、pushwindow到下个页面)，会触发此事件。

备注：10.0.15客户端以上，容器新增了pagePause和appPause事件，用于业务区分是哪种情况触发的pause。

注意： 由于Android原生的resume和pause事件不能区分是压后台导致还是页面切换导致，所以pageResume和pagePause事件是通过jsapi调用记录去回调的，所以仅适用于同一个session内window之间的互相切换。对于startApp和其他客户端直接切换页面方式（比如chooseImage)不生效。

## 1. 使用方法

```
document.addEventListener('pause', function (e) {
    alert("pause");
}, false);
```

## 2. 代码演示

### 2.1. 离开当前页面后弹出alert

```
<h1>请点击按钮打开一个新窗口</h1>
<a href="javascript:void(0)" class="btn btn-default J_new_window">新窗口打开当前页面</a>
<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function() {
    document.querySelector('.J_new_window').addEventListener('click', function() {
        AlipayJSBridge.call('pushWindow', {
            url: location.pathname,
        });
    });
});

document.addEventListener('pause', function (e) {
    alert('paused');
}, false);
</script>
```

Since 9.9

# popMenuClick

标题栏右边popMenu点击后触发的事件

## 1. 使用方法

```
document.addEventListener('popMenuClick', function (e) {  
    // 得到name属性值  
    console.log(e.data.title); // 这里不是name, 是title  
}, false);
```

## 2. 代码演示

### 2.1. 基本功能演示

```
<h1>请点击右上角弹出的菜单</h1>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  AlipayJSBridge.call('showPopMenu',{
    menus:[
      {name:"字体", tag:"tag1"},
      {name:"菜单项2", tag:"tag1"},
      {name:"菜单项3", tag:"tag1"}
    ]
  }, function (e) {
    console.log(e);
  });
});

document.addEventListener('popMenuClick', function (e) {
  alert(JSON.stringify(e.data));
}, false);
</script>
```

Since 8.1

# resume

当一个webview界面重新回到栈顶时(从后台被唤起、锁屏界面恢复、从下个页面回退)，会触发此事件。  
如果这个界面是通过popWindow/popTo到达，且传递了data参数，此页可以获取到data。  
如果在界面的resume之前先发生了app的resume，则event还会有一个resumeParams，包含app resume时接收到的参数

备注：10.0.15客户端以上，容器新增了pageResume和appResume事件，用于业务区分是哪种情况触发的resume。  
注意： 由于Android原生的resume和pause事件不能区分是压后台导致还是页面切换导致，所以pageResume和pagePause事件是通过jsapi调用记录去回调的，所以仅适用于同一个session内window之间的互相切换。对于startApp和其他客户端直接切换页面方式（比如chooseImage)不生效。

# 1. 使用方法

```
document.addEventListener('resume', function (e) {
  console.log("resumed");
}, false);
```

## 2. 代码演示

### 2.1. 返回的时候有数据带过来

```
<h1>首先打开一个页面，新页面返回时带数据回这个页面</h1>
<a href="#" class="btn btn-default J_demo">打开新页面</a>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('a').addEventListener('click', function(){
    AlipayJSBridge.call('pushWindow', {
      url: '/jsapi/context/pop-window-data.html'
    });
  });
});

document.addEventListener('resume', function (event) {
  alert('页面回退时带过来的内容:  ' + JSON.stringify(event.data));
});
</script>
```

## 3. API

### 3.1. 出参

事件处理函数的event会有2个属性

| 名称 | 类型 | 描述 |
|----|----|----|
|    |    |    |



|              |        |                        |
|--------------|--------|------------------------|
| data         | object | popTo或者popWindow带过来的内容 |
| resumeParams | object | 包含app resume时接收到的参数    |

Since 8.2

# subtitleClick

点击导航栏副标题触发回调

## 1. 使用方法

```
document.addEventListener('subtitleClick', function (e) {  
  alert('subtitle clicked')  
}, false);
```

## 2. 代码演示

### 2.1. 基本功能演示

```
<h1>请点击副标题查看效果</h1>
<script>
document.addEventListener('subtitleClick', function (e) {
    alert('subtitle clicked')
}, false);

function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    AlipayJSBridge.call('setTitle', {
        title: '标题',
        subtitle: '副标题'
    });
});
</script>
```

Since 8.2

# titleClick

点击导航栏标题触发回调

## 1. 使用方法

```
document.addEventListener('titleClick', function (e) {
    alert('title clicked')
}, false);
```

## 2. 代码演示

### 2.1. 基本功能演示

<h1>请点击标题查看效果</h1>

```
<script>
document.addEventListener('titleClick', function (e) {
  alert('title clicked')
}, false);
</script>
```

# 应用简述

## 1. 接口通用调用示例

```
// 由于机制问题，JSAPI注入是异步过程，时序无法保证。
// 如果使用原生AlipayJSBridge接口，需要先监听AlipayJSBridgeReady事件后再调用原生JSAPI接口。
// 建议使用antBridge.js来调用JSAPI，无需考虑时序问题。
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function () {
  AlipayJSBridge.call('toast', {
    content: 'HELL0'
  });
});
```

## 2. 使用AlipayJSAPI

- alipayjsapi (../alipayjsapi/)

## 3. 错误处理约定

jsAPI的回调函数的参数是一个**dictionary**，里面可能包含一个特殊的字段error，作为api调用的错误码，示例

```
{
  error: 1,
  errorMessage: '接口不存在'
}
```

小于10为通用错误码， 具体api对应使用。定义以下通用错误码:

| error | 含义     |
|-------|--------|
| 1     | 接口不存在  |
| 2     | 参数无效   |
| 3     | 发生未知错误 |
| 4     | 接口无权限  |

## 4. 界面名词简介

- 图片中加红框的部分， 均是**native**界面， 可由**AlipayJSBridge**的**api**控制显示方式
- Native界面部分不占用window窗口高度， css中无需预留高度

### 4.1. 标题栏



## 5. 环境变量

### 5.1. navigator.userAgent

包含钱包版本信息，如

```
"Mozilla/5.0.....AlipayDefined(nt:WIFI,ws:320|504|2.0) AliApp(AP/9.6.0.000001) AlipayClient/9.6.0.000001 Language/zh-Hans ProductType/dev"
```

## 5.2. AlipayJSBridge.startupParams

可通过该环境变量获取当前app的启动参数，如

```
{"appId":"xxxxxxx","st": "YES", "sb": "NO", "url": "https://m.taobao.com/"}
```

**警告：** 由于iOS容器的某个缓存机制兼容性问题，极端场景下通过**startupParams**环境变量可能会获取到上一次打开相同页面的值，因此切勿通过启动参数机制来传递重要的自定义业务参数。

# chooseImage

拍照或从手机相册中选图接口

## 1. 使用方法

```
AlipayJSBridge.call('chooseImage', {
  sourceType: ['camera', 'album'],
  count: 9
}, function (result) {
  console.log(result);
});
```

## 2. 代码演示

### 2.1. 基本使用

```

<h1>点击按钮选择图片</h1>
<button id="J_chooseImage">选择图片</button>
<script type="text/javascript">
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function() {
    var btn = document.getElementById('J_chooseImage');

    btn.addEventListener('click', function(e) {
        AlipayJSBridge.call(
            'chooseImage',
            {
                count: 1
            },
            function (result) {
                alert('chooseImage: '+JSON.stringify(result));
                var uploadParams = {
                    url: 'http://h5test.inc.alipay.net:3000/upload',
                    name: 'h5test'
                };
                if(result.localIds && result.localIds.length > 0){
                    uploadParams.localId = result.localIds[0];
                    uploadParams.type = 'image';
                } else if(result.tempFilePaths && result.tempFilePaths.length > 0){
                    uploadParams.filePath = result.tempFilePaths[0].replace('file://', '');
                }
                AlipayJSBridge.call('uploadFile', uploadParams, function (result) {
                    alert('uploadFile: '+JSON.stringify(result));
                });
            }
        );
    }, false);
});
</script>

```

### 3. API

```
AlipayJSBridge.call('chooseImage', {
  sourceType: 'camera',
  count: 1
}, fn);
```

### 3.1. 入参

| 名称           | 类型     | 描述                              | 必选 | 默认值                   | 版本    |
|--------------|--------|---------------------------------|----|-----------------------|-------|
| count        | number | 最大可选照片数                         | N  | 默认9张                  | 9.9.7 |
| sourceType   | array  | 相册选取或者拍照                        | N  | [ ‘camera’, ’album’ ] | 9.9.7 |
| publicDomain | bool   | 是否准备上传到公有域，上传到公有域的图片在访问的时候不需要鉴权 | N  | false                 |       |

### 3.2. 出参

|                           |
|---------------------------|
| 回调函数带入的参数result: {error } |
|---------------------------|

| 名称           | 类型         | 描述                 |
|--------------|------------|--------------------|
| localIds     | array      | 图片多媒体LocalId iOS使用 |
| tempFilePath | jsonString | 图片文件路径 Android使用   |
| success      | bool       | 是否返回成功             |
| errorCode    | string     | 错误码                |

### 3.3. 错误

| errorCode | 描述         |
|-----------|------------|
| 10        | 用户取消       |
| 11        | 操作失败（权限不够） |

## 4. 使用注意

- 获取localID后，如果使用uploadImage接口上传，此处publicDomain参数的值应和uploadImage接口的publicDomain参数值的保持一致。



# compressImage

压缩图片接口(安卓10.0.12版本支持, ios 10.0.15版本支持)

## 1. 使用方法

```
AlipayJSBridge.call('compressImage', {
  apFilePaths: ["https://resource/apmlcc0ed184daffc5a0d8da86b2f518cf7b.image"],
  compressLevel:4
}, function(result){
  console.log(result);
});
```

## 2. 代码演示

### 2.1. 基本使用

```
<h1>点击按钮选择图片</h1>
<button id="J_chooseImage">选择图片</button>
<script type="text/javascript">
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function() {
    var btn = document.getElementById('J_chooseImage');

    btn.addEventListener('click', function(e) {
        AlipayJSBridge.call(
            'chooseImage',
            {
                count: 1
            },
            function (result) {
                alert('chooseImage: '+JSON.stringify(result));
                var uploadParams = {
                    url: 'http://h5test.inc.alipay.net:3000/upload',
                    name: 'h5test'
                };
                if(result.localIds && result.localIds.length > 0){
                    uploadParams.localId = result.localIds[0];
                    uploadParams.type = 'image';
                } else if(result.tempFilePaths && result.tempFilePaths.length > 0){
                    uploadParams.filePath = result.tempFilePaths[0].replace('file://', '');
                }
                AlipayJSBridge.call('uploadFile', uploadParams, function (result) {
                    alert('uploadFile: '+JSON.stringify(result));
                });
            }
        );
    }, false);
});
</script>
```

### 3. API

```
AlipayJSBridge.call('compressImage', {
  apFilePaths:[filePaths],
  compressLevel:int
}, function(result){
  console.log(result);
});
```

3.1. 入参

| 名称            | 类型    | 描述        | 必选 | 默认值 | 版本      |
|---------------|-------|-----------|----|-----|---------|
| apFilePaths   | array | filePaths | Y  |     | 10.0.12 |
| compressLevel | int   | 压缩质量      | Y  | [4] | 10.0.12 |

3.2. 压缩质量

| compressLevel | 描述     |
|---------------|--------|
| 0             | 低质量    |
| 1             | 中等质量   |
| 2             | 高质量    |
| 3             | 不压缩    |
| 4             | 根据网络适应 |

3.3. 出参

|                           |
|---------------------------|
| 回调函数带入的参数result: {error } |
|---------------------------|

| 名称          | 类型    | 描述           |
|-------------|-------|--------------|
| apFilePaths | array | 压缩后的filePath |

3.4. 错误

| errorCode | 描述 |
|-----------|----|
|           |    |

# startAudioRecord

录制语音文件

## 1. 使用方法

```
AlipayJSBridge.call('startAudioRecord', {
  maxRecordTime: 60, // 最大录音时长, 默认60s
  minRecordTime: 1, // 最小录音时长, 默认1s
  business: 'apm-h5', // 业务标识
}, function (result) {
  console.log(result);
});
```

## 2. 代码演示

### 2.1. 基本使用

<h1>点击按钮开始录音</h1>

```
<button id="J_recordStart" class="btn btn-default read">开始录音</button>
<button id="J_recordStop" class="btn btn-default read">停止录音</button>
<button id="J_recordPlay" class="btn btn-default read">播放录音</button>
<script type="text/javascript">
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
```

```
  var btnStart = document.getElementById('J_recordStart');
  var btnStop = document.getElementById('J_recordStop');
  var btnPlay = document.getElementById('J_recordPlay');
```

```

var localId;

btnStart.addEventListener('click',function(e){
  alert('开始录音');
  AlipayJSBridge.call('setTitle', {
    subtitle: '正在录音...'
  });
  AlipayJSBridge.call('startAudioRecord', {
    maxRecordTime: 60, // 最大录音时长, 默认60s
    minRecordTime: 1, // 最小录音时长, 默认1s
    business: 'apm-h5', // 业务标识
  }, function (result) {
    alert('startAudioRecord: '+JSON.stringify(result));
    localId = result.identifier;
  });
});

btnStop.addEventListener('click',function(e){
  alert('停止录音');
  AlipayJSBridge.call('stopAudioRecord', {
  }, function (result) {
    alert('stopAudioRecord: '+JSON.stringify(result));
    AlipayJSBridge.call('setTitle', {
      subtitle: ''
    });
  });
});

btnPlay.addEventListener('click',function(e){
  if(!localId){
    alert('先录一段吧');
    return;
  }
  alert('播放录音');
  AlipayJSBridge.call('startPlayAudio', {
    identifier: localId, // 音频文件id, 可以是cloudid 或者 本地id
    business: 'apm-h5', // 业务标识
  }, function (result) {
    alert('startPlayAudio: '+JSON.stringify(result));
  });
});

});
</script>

```

### 3. API

```
AlipayJSBridge.call('startAudioRecord', {
  maxRecordTime: 60, // 最大录音时长, 默认60s
  minRecordTime: 1, // 最小录音时长, 默认1s
  business: 'apm-h5', // 业务标识
}, function (result) {
  console.log(result);
});
```

3.1. 入参

| 名称            | 类型       | 描述         | 必选 | 默认值 | 版本     |
|---------------|----------|------------|----|-----|--------|
| maxRecordTime | number   | 最大录音时长 单位s | N  | 60  | 10.0.3 |
| minRecordTime | number   | 最小录音时长 单位s | N  | 1   | 10.0.3 |
| business      | string   | 业务标识       | Y  |     | 10.0.3 |
| fn            | function | 回调函数       | N  |     |        |

3.2. 出参

回调函数会在stopAudioRecord接口调用后执行

| 名称           | 类型     | 描述                |
|--------------|--------|-------------------|
| identifier   | string | 录音多媒体唯一标记 localId |
| tempFilePath | string | 临时文件路径            |
| duration     | number | 录制时长              |
| errorMessage | string | 错误描述              |
| error        | string | 错误码               |

3.3. 错误

| errorCode | 描述 |
|-----------|----|
|           |    |

Since 9.9.0

# beehiveOptionsPicker

类似于safari原生select的组件，但是功能更加强大 一般用来替代select，或者2级数据的选择，注意**不支持2级数据之间的联动**。

## 1. 使用方法

```
// 单列选项
AlipayJSBridge.call('beehiveOptionsPicker', {
  title:"hello",
  optionsOne:["1","2","3"]
}, function (result) {
  console.log(result);
});

// 双列选项
AlipayJSBridge.call('beehiveOptionsPicker',{
  title:"hello",
  optionsOne:["1","2","3"],
  optionsTwo:["1","2","3"]
},function(result){
  console.log(result);
});
```

## 2. 代码演示

### 2.1. 单列选择功能

```
<h1>点击按钮查看效果</h1>
<a href="javascript:void(0)" class="invoke">选择还款日</a>

<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function() {
  document.querySelector('.invoke').addEventListener('click', function(){
    AlipayJSBridge.call('beehiveOptionsPicker', {
      title:"还款日选择",
      optionsOne:["每周一","每周二","每周三","每周四","每周五","每周六","每周日"],
      selectedOneIndex: 2,
    }, function(result) {
      alert(JSON.stringify(result));
    });
  })
}, false);
</script>
```

2.2. 双列选择功能



<h1>点击按钮查看效果</h1>  
<a href="javascript:void(0)" class="invoke">选择出生年月</a>

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function() {
  document.querySelector('.invoke').addEventListener('click', function(){
    AlipayJSBridge.call('beehiveOptionsPicker', {
      title:"出生年月选择",
      optionsOne:["2014年","2013年","2012年","2011年","2010年","2009年","2008年"],
      optionsTwo:["一月",'二月','三月','四月','五月','六月','七月','八月','九月','十月','十一月','十二月'],
      selectedOneIndex: 3,
      selectedTwoIndex: 5,
    }, function(result) {
      alert(JSON.stringify(result));
    });
  });
}, false);
</script>
```

### 3. API

```
AlipayJSBridge.call('beehiveOptionsPicker', {
  title, optionsOne, optionsTwo, selectedOneIndex,
  selectedTwoIndex, positiveString, negativeString
}, fn)
```

#### 3.1. 入参

| 名称         | 类型     | 描述     | 必选 | 默认值 | 版本  |
|------------|--------|--------|----|-----|-----|
| title      | string | 头部标题信息 | N  |     | 9.9 |
| optionsOne | Array  | 选项一列表  | Y  |     |     |
| optionsTwo | Array  | 选项二列表  | N  |     |     |
|            |        |        |    |     |     |

|                  |          |         |   |      |  |
|------------------|----------|---------|---|------|--|
| selectedOneIndex | number   | 选项一默认选中 | N | 0    |  |
| selectedTwoIndex | number   | 选项二默认选中 | N | 0    |  |
| positiveString   | string   | 确定按钮文案  | N | ‘确定’ |  |
| negativeString   | string   | 取消按钮文档  | N | ‘取消’ |  |
| fn               | function | 回调函数    | N |      |  |

3.2. 出参

|   |
|---|
| 回调函数带入的参数result: {selectedOneIndex, selectedOneOption, selectedTwoIndex, selectedTwoOption} |
|---|

| 名称                | 类型     | 描述       | 备注 |
|-------------------|--------|----------|----|
| selectedOneIndex  | number | 选项一选择的值  |    |
| selectedOneOption | string | 选项一选择的内容 |    |
| selectedTwoIndex  | number | 选项二选择的值  |    |
| selectedTwoOption | string | 选项二选择的内容 |    |

Since 8.3

# chooseContact

唤起支付宝通讯录，选择一个或者多个支付宝联系人

## 1. 使用方法

```
AlipayJSBridge.call('chooseContact', {
  "type": "single", // "single" or "multi", default multi
  "title": "choose contacts", // title show on the title bar
  "multiMax": 20, // max contact items size, default 50
  "multiMaxText": "max!", // message if selected more than max
}, function(result) {
  // error code:
  // 10: no permission
  // 11: user canceled
  console.log(result.error);

  // contacts as JSON array, for each item contains the following fields:
  // "name": "dawsondong",
  // "mobile": 18666666666,
  // "email": h5@alipay.com,
  // "userId": "10280023927349823798",
  // "headImageUrl": "head.alipay.com/xx.png",
  // "from": "account" // "account","contact", "input"
  console.log(JSON.stringify(result.contacts));
});
```

## 2. 代码演示

### 2.1. 基本功能

<h1>点击选择多个联系人</h1>  
<a href="javascript:void(0)" class="contact">打开手机通信录</a>

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.contact').addEventListener('click', function(){
    AlipayJSBridge.call('chooseContact', {
      "title": "choose contacts", // title show on the title bar
      "multiMax": 2, // max contact items size, default 50
      "multiMaxText": "max!", // message if selected more than max
    }, function(result) {
      alert(JSON.stringify(result));
    });
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('chooseContact', {
  title, type, multiMax, multiMaxText
}, fn)
```

#### 3.1. 入参

| 名称           | 类型       | 描述                         | 必选 | 默认值    | 版本 |
|--------------|----------|----------------------------|----|--------|----|
| title        | string   | 显示在联系人选择页面的标题              | N  | 跟随系统默认 |    |
| type         | string   | 选择模式，多选(multi)或者单选(single) | N  | multi  |    |
| multiMax     | int      | 单次最多选择联系人数（0或-1，表示无限制选择）   | N  | 50     |    |
| multiMaxText | string   | 用户选择数超过上限的文案               | N  | 跟随系统默认 |    |
| fn           | function | 回调函数                       | N  |        |    |

3.2. 出参

回调函数带入的参数result: {contacts: {email, mobile, headImageUrl, name, userId, from}}

| 名称           | 类型     | 描述        |
|--------------|--------|-----------|
| email        | string | 账号的邮箱     |
| mobile       | string | 账号对应的手机号码 |
| headImageUrl | string | 账号的头像     |
| name         | string | 账号的真实姓名   |
| from         | string | 账号来源      |

3.3. 错误

| error | 描述     |
|-------|--------|
| 10    | 操作无权限； |
| 11    | 用户取消操作 |

Since 8.0

# contact

选择本地系统通信录中某个人的一个联系电话

1. 使用方法

```
AlipayJSBridge.call("contact", function(result) {
  console.log(result.name, result.mobile);
});
```

## 2. 代码演示

### 2.1. 基本功能

```
<h1>点击调出本地通讯录</h1>
<p>请注意要在界面中选择2级</p>
<a href="javascript:void(0)" class="contact">打开通信录</a>

<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.contact').addEventListener('click', function(){
    AlipayJSBridge.call("contact", function(result) {
      alert(JSON.stringify(result))
    });
  });
});
</script>
```

## 3. API

AlipayJSBridge.call('contact', fn)

### 3.1. 入参

| 名称 | 类型       | 描述   | 必选 | 默认值 | 版本 |
|----|----------|------|----|-----|----|
| fn | function | 回调函数 | N  |     |    |

### 3.2. 出参

|                                 |
|---------------------------------|
| 回调函数带入的参数result: {name, mobile} |
|---------------------------------|

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|--|--|--|--|

| 名称     | 类型     | 描述        | 备注 |
|--------|--------|-----------|----|
| name   | string | 选中的联系人姓名  |    |
| mobile | string | 选中的联系人手机号 |    |

### 3.3. 错误

| error | 描述                   |
|-------|----------------------|
| 10    | 没有权限；                |
| 11    | 用户取消操作（或设备未授权使用通讯录）； |

# datePicker

日期选择或者时间选择

## 1. 使用方法

```
AlipayJSBridge.call('datePicker',{
  beginDate: '2016-10-10',
  minDate: '2016-10-9',
  maxDate: '2017-10-9',
  mode:1,
}, function(e){
  alert(JSON.stringify(e))
})

AlipayJSBridge.call('datePicker',{
  minDate: '2015-1-1 0:0:0',
  maxDate: '2050-1-1 0:0:0',
  mode: 2
},function(e) {
  console.log(e);
});
```

## 2. 代码演示

## 2.1. 不同模式下的日期/时间选择

<h1>点击按钮查看各个模式下日期选择</h1>

<a href="javascript:void(0)" class="btn btn-default date">选择日期</a>  
<a href="javascript:void(0)" class="btn btn-default time">选择时间</a>  
<a href="javascript:void(0)" class="btn btn-default datetime">选择日期+时间</a>

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.date').addEventListener('click', function(){
    AlipayJSBridge.call('datePicker',{
      beginDate: '2016-10-10',
      minDate: '2016-10-9',
      maxDate: '2017-10-9',
      mode:1,
    }, function(e){
      alert(JSON.stringify(e))
    })
  });
  document.querySelector('.time').addEventListener('click', function(){
    AlipayJSBridge.call('datePicker',{
      mode:0,
    }, function(e){
      alert(JSON.stringify(e))
    })
  });
  document.querySelector('.datetime').addEventListener('click', function(){
    AlipayJSBridge.call('datePicker',{
      minDate: '2016-10-9',
      maxDate: '2017-10-9',
      beginDate: '2016-10-9',
      mode:2,
    }, function(e){
      alert(JSON.stringify(e))
    })
  });
});
</script>
```



2.2. 身份证日期选择

```
<h1>点击按钮选择身份证有效期</h1>

<a href="javascript:void(0)" class="btn btn-default date">选择日期</a>

<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.date').addEventListener('click', function(){
    AlipayJSBridge.call('datePicker',{
      beginDate: '2015-10-12',
      minDate: '2015-10-09',
      maxDate: '2016-10-09',
      mode:1,
      isIDCard: true,
    }, function(e){
      alert(JSON.stringify(e))
    })
  });
});
</script>
```

3. API

```
AlipayJSBridge.call('datePicker', {
  mode, beginDate, minDate, maxDate, isIDCard
}, fn);
```

3.1. 入参

| 名称        | 类型     | 描述   | 必选 | 默认值 | 版本 |
|-----------|--------|--|----|-----|----|
| mode      | number | 0: HH:mm:ss, 1: yyyy-MM-dd, 2: yyyy-MM-dd HH:mm:ss | N  | 1   |    |
| beginDate | string | 默认选择的日期/时间，不设置默认为当前系统时间                            | N  | ‘’  |    |
|           |        |  |    |     |    |

|          |          |                       |   |    |  |
|----------|----------|-----------------------|---|----|--|
| minDate  | string   | 最小日期/时间，必须小于beginDate | N | “  |  |
| maxDate  | string   | 最大日期/时间，必须大于beginDate | N | ”  |  |
| isIDCard | bool     | 是否是身份证模式              | N |    |  |
| fn       | function | 回调函数                  | N | 不限 |  |

### 3.2. 出参

|                                |
|--------------------------------|
| 回调函数带入的参数result: {error, date} |
|--------------------------------|

| 名称    | 类型     | 描述           | 版本 |
|-------|--------|--------------|----|
| error | int    | 错误码（11：用户取消） |    |
| date  | string | 选择的日期/时间字符串  |    |

## 4. 使用注意

- mode虽然以日期格式字符串表示，但是并不是说返回的格式是这样的，返回的格式是根据系统格式决定的，比如返回日期可能是2016-10-10。也就是说，它的作用是决定选择什么。
- 如果要通过代码隐藏日期选择，请使用 AlipayJSBridge.call('hideDatePicker');
- minDate，maxDate，beginDate 并不只是日期，还可以是时间。
- isIDCard 的作用是让我们可以选择**长期**，英文系统下是\*\*Long Term，公民在46周岁后领取身份证，有效期是长期。
- 特别注意，安卓下传入参数的时候，必须做到 beginDate >= minDate 以及 beginDate <= maxDate，否则会报参数错误。

Since 8.1

# getCities

getCities用来呼出城市选择列表

## 1. 使用方法

```
AlipayJSBridge.call('getCities', function (result) {
    console.log(result);
});
```

## 2. 代码演示

### 2.1. 基本使用演示

```
<h1>点击打开选择城市</h1>
<a href="#" class="btn btn-default J_demo">打开城市列表</a>
<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('a').addEventListener('click', function(){
        AlipayJSBridge.call('getCities', function (result) {
            alert(JSON.stringify(result));
        });
    });
});
</script>
```

### 2.2. 城市选择并开启定位功能

```
<h1>点击打开选择城市</h1>
<a href="#" class="btn btn-default J_demo">打开城市列表</a>
<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('a').addEventListener('click', function(){
        // 需要同时传入城市名称和adcode才能开启定位功能
        AlipayJSBridge.call('getCities', {
            currentCity: '杭州市', // 当前选择的城市, 一般传入上次选择的就可以
            adcode: '310100'
        }, function (result) {
            alert(JSON.stringify(result));
        });
    });
});
</script>
```

### 2.3. 自定义城市列表，并且不显示热门城市

```
<h1>点击打开选择城市</h1>
<a href="#" class="btn btn-default J_demo">打开城市列表</a>
<script>
  function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
      callback && callback();
    } else {
      // 如果没有注入则监听注入的事件
      document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
  }
  ready(function(){
    document.querySelector('a').addEventListener('click', function(){
      AlipayJSBridge.call('getCities', {
        needHotCity:false,
        customCities:[{name:"杭州",adcode:"xxxx",pinyin:"hangzhou"},{name:"杭州",adcode:"xxxx",pinyin:"hangzhou"}]
      }, function (result) {
        alert(JSON.stringify(result));
      });
    });
  });
</script>
```

### 3. API

```
AlipayJSBridge.call('getCities',{
  currentCity,
  adcode,
  needHotCity,
  customCities
}, fn)
```

#### 3.1. 入参

| 名称              | 类型     | 描述  | 必选 | 默认值 | 版本  |
|-----------------|--------|---|----|-----|-----|
| currentCity     | string | 传入当前使用中的城市名   | N  |     | 8.2 |
| adcode          | string | 传入当前使用中的城市的adcode（如果没有用到adcode，可以传入空字符串）                | N  |     | 8.2 |
| customCities    | array  | 自定义城市列表（格式：{name:”杭州”,adcode:”xxxx”,pinyin:”hangzhou”}） | N  |     | 8.4 |
| customHotCities | array  | 自定义热门城市列表   | N  |     | 8.4 |
|                 |        |   |    |     |     |

|             |          |             |   |      |     |
|-------------|----------|-------------|---|------|-----|
| needHotCity | bool     | 是否需要现实热门城市  | N | true | 8.4 |
| fn          | function | 当选择了城市之后的回调 |   |      |     |

### 3.2. 出参

|                                 |
|---------------------------------|
| 回调函数带入的参数result: {city, adcode} |
|---------------------------------|

## 4. 使用注意

- getCities 如果要开启定位，那么需要同时传入 currentCity 和 adcode

Since 8.1

# getLocation

getLocation用来获取用户当前的地理位置信息

## 1. 使用方法

```
AlipayJSBridge.call('getLocation', function (result) {
  alert(JSON.stringify(result));
});
```

## 2. 代码演示

### 2.1. 获取用户详细地理位置信息

```
<h1>点击获取详细地理位置信息</h1>
<p>一般情况下还会返回周围兴趣点</p>
<a href="#" class="btn btn-default read">点击获取定位信息</a>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.read').addEventListener('click', function(){
    AlipayJSBridge.call('getLocation', function (result) {
      if (result.error) {
        alert(result.errorMessage);
        return;
      }

      alert(JSON.stringify(result));
      console.log(result.province, result.city, result.latitude, result.longitude, result.citycode, result.adcode, result.streetNumber, result.p
ois);
      // 四个直辖市（北京, 上海, 重庆, 天津）返回的city为空字符串, 可以通过result.province来获取, 所以建议result.city?result.city:result.province这样来获取city
    });
  });
});
</script>
```

## 2.2. 获取用户地址经纬度

```
<h1>点击获取用户所在地经纬度</h1>
<a href="#" class="btn btn-default read">点击获取定位信息</a>
<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('.read').addEventListener('click', function(){
        AlipayJSBridge.call('getLocation', {requestType: 2}, function (result) {
            if (result.error) {
                alert(result.errorMessage);
                return;
            }

            alert(JSON.stringify(result));
        });
    });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('getLocation', {
    cacheTimeout, requestType, reGeocodeServiceType, timeout, horizontalAccuracy
}, fn)
```

| 名称                 | 类型       | 描述  | 必选 | 默认值 | 版本    |
|--------------------|----------|---|----|-----|-------|
| cacheTimeout       | float    | 钱包经纬度定位缓存过期时间，单位秒，例如：钱包上一次定位时间是10:00:00，当前调用cacheTimeout=5，当前时间是10:00:06，则判断缓存过期不可用，会重新定位 | N  | 30  | 9.0   |
| requestType        | int      | 0：详细逆地理编码，带周边信息（当使用MobileLBS服务时，当遇到流量巨大时可能会做降级处理，届时可能没有poi周边信息）<br>1：逆地理编码到城市<br>2：仅获取经纬度 | N  | 0   | 9.0.1 |
| timeout            | float    | 定位超时失败回调时间，单位秒，默认是10s   | N  | 10  | 9.0.1 |
| horizontalAccuracy | float    | 定位精度，默认是不限制精度。当超过误差精度时，会继续等待位置更新，直到超时，超时处理使用最后一次定位的经纬度。                                   | N  |     | 9.0.1 |
| fn                 | function | 定位完成后被调用的回调函数   | N  |     | 9.0.1 |



### 3.1. 出参

回调函数带入的参数result: {error, extError, latitude, longitude, country, countryCode, city, province, citycode, adcode, streetNumber, pois}

| 名称           | 类型     | 描述  | 版本    |
|--------------|--------|---|-------|
| error        | int    | 错误码（12：GPS打开，但定位失败；13：获取地理位置信息失败；14：定位超时；15：网络错误；16:GPS未打卡或未授权） | 8.6   |
| extError     | int    | 扩展错误码（android上由高德SDK导出）   | 8.6   |
| latitude     | double | 纬度  | 8.1   |
| longitude    | double | 经度  | 8.1   |
| country      | string | 国家(requestType为0, 1时生效)   | 9.6.6 |
| countryCode  | string | 国家编号(requestType为0, 1时生效)                                       | 9.6.6 |
| city         | string | 城市，建议result.city?result.city:result.province这样来获取city)          | 8.1   |
| province     | string | 省份  | 8.1   |
| citycode     | string | 城市编码  | 8.2   |
| adcode       | string | 区域编码  | 8.2   |
| streetNumber | object | 门牌信息，结构是：{street, number}                                       | 8.2   |
| pois         | array  | 周边兴趣点   | 8.2   |

### 4. 使用注意

- getLocation 不能获取provinceCode。
- getLocation 不能获取district信息，比如 中国浙江省杭州市西湖区万塘路18号 这个地址， getLocation 不会返回**西湖区**这个信息。

Since 8.0

## getNetworkType

获取网络状态: 网络是否可用以及什么网络类型

## 1. 使用方法

```
AlipayJSBridge.call('getNetworkType', function (result) {
  console.log(result.networkType);
});
```

## 2. 代码演示

### 2.1. 基本功能演示

```
<h1>点击获取网络信息，可以修改网络后再点击看看</h1>
<a href="#" class="btn btn-default J_demo">查看网络信息</a>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('a').addEventListener('click', function(){
    AlipayJSBridge.call('getNetworkType', function (result) {
      alert(JSON.stringify(result));
    });
  });
});
</script>
```

## 3. API

```
AlipayJSBridge.call('getNetworkType', fn)
```

### 3.1. 入参

| 名称 | 类型 | 描述 | 必选 | 默认值 | 版本 |
|----|----|----|----|-----|----|
|    |    |    |    |     |    |

|    |          |      |  |     |
|----|----------|------|--|-----|
| fn | function | 回调函数 |  | 8.0 |
|----|----------|------|--|-----|

### 3.2. 出参

|                                   |
|-----------------------------------|
| 回调函数带入的参数result，以networkInfo返回值为准 |
|-----------------------------------|

| 名称          | 类型     | 描述  | 版本  |
|-------------|--------|---|-----|
| networkInfo | string | 网络类型值：UNKNOWN/NOTREACHABLE/WIFI/3G/2G/4G/WWAN | 9.6 |

Since 8.6

# imageViewer

imageViewer是一个native实现的图片查看组件

## 1. 使用方法

```
AlipayJSBridge.call('imageViewer', {
  images: [{
    u: 'http://a.hiphotos.baidu.com/image/pic/item/1b4c510fd9f9d72a80f8481ad62a2834349bbb1d.jpg',
    t: 'http://a.hiphotos.baidu.com/image/pic/item/1b4c510fd9f9d72a80f8481ad62a2834349bbb1d.jpg'
  }],
  init: 0
}, function(result) {
  console.log(result);
});
```

## 2. 代码演示

### 2.1. 基本功能

<h1>点击查看图片预览效果</h1>

```
<a href="#" class="btn btn-default read">开始预览</a>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.read').addEventListener('click', function(){
    AlipayJSBridge.call('imageViewer',
      {
        images: [
          {u: 'http://a.hiphotos.baidu.com/image/pic/item/1b4c510fd9f9d72a80f8481ad62a2834349bbb1d.jpg', t: ''},
          {u: 'http://a.hiphotos.baidu.com/image/pic/item/1b4c510fd9f9d72a80f8481ad62a2834349bbb1d.jpg', t: ''}
        ],
        init: 0
      },
      function(result) {
        console.log(result);
      });
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('imageViewer', {
  images: {u, t}, init
}, fn);
```

| 名称     | 类型     | 描述      | 必选 | 默认值 | 版本 |
|--------|--------|---------|----|-----|----|
| images | array  | 图片数组    | Y  |     |    |
| u      | string | 图片url   | N  | “   |    |
| t      | string | 缩略图url  | N  |     |    |
| init   | int    | 默认显示第几张 | N  |     |    |
|        |        |         |    |     |    |

|    |          |            |   |  |  |
|----|----------|------------|---|--|--|
| fn | function | API调用结果的回调 | N |  |  |
|----|----------|------------|---|--|--|

### 3.1. 出参

|                                   |
|-----------------------------------|
| 回调函数带入的参数result: {error, success} |
|-----------------------------------|

Since 8.1

# photo

拍照/选择照片

## 1. 使用方法

```
AlipayJSBridge.call('photo', {
  dataType: 'dataURL',
  imageFormat: 'jpg',
  quality: 75,
  maxWidth: 500,
  maxHeight: 500,
  allowEdit: true,
  multimediaConfig: { // 可选, 仅当该项被配置时, 图片被传输至 APMultimedia
    compress: 2, // 可选, 默认为4。 0-低质量, 1-中质量, 2-高质量, 3-不压缩, 4-根据网络情况自动选择
    business: "multiMedia" // 可选, 默认为“NebulaBiz”
  }
}, function (result) {
  alert(JSON.stringify(result))
});
```

## 2. 代码演示

### 2.1. 基本功能

<h1>点击查看选择一幅图片</h1>

```
<a href="#" class="btn btn-default read">调用图片选择</a>
<img id="myImage" />
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.read').addEventListener('click', function(){
    AlipayJSBridge.call('photo', {
      dataType: 'dataURL',
      imageFormat: 'jpg',
      quality: 75,
      maxWidth: 500,
      maxHeight: 500,
      allowEdit: true,
      multimediaConfig: { // 可选, 仅当该项被配置时, 图片被传输至 APMultimedia
        compress: 2, // 可选, 默认为4。 0-低质量, 1-中质量, 2-高质量, 3-不压缩, 4-根据网络情况自动选择
        business: "multiMedia" // 可选, 默认为"NebulaBiz"
      }
    }, function (result) {
      var image = document.getElementById('myImage');
      image.src = "data:image/jpeg;base64," + result.dataURL;
    });
  });
});
</script>
```

3. API

```
AlipayJSBridge.call('photo', {
  dataType,
  imageFormat,
  quality,
  maxWidth,
  maxHeight,
  allowEdit,
  multimediaConfig: { // 可选, 仅当该项被配置时, 图片被传输至 APMultimedia
    compress, // 可选, 默认为4。 0-低质量, 1-中质量, 2-高质量, 3-不压缩, 4-根据网络情况自动选择
    business // 可选, 默认为“NebulaBiz”
  }
}, fn);
```

3.1. 入参

| 名称               | 类型         | 描述   | 必选 | 默认值       | 版本             |
|------------------|------------|--|----|-----------|----------------|
| dataType         | string     | 结果数据格式：<br>dataURL：base64编码的图片数据<br>fileURL：图片在文件系统中的url<br>图片存放于临时目录中，钱包退出时被清除  | Y  | ‘dataURL’ |                |
| imageFormat      | string     | jpg / png  | N  | ‘jpg’     |                |
| quality          | int        | jpg的图片质量，取值1到100   | N  | 75        |                |
| beautyLevel      | float      | 是否返回美颜后的图片，取值0~1的浮点值，代表美颜效果的程度   | N  | 0         | 9.6            |
| maxWidth         | int        | 图片的最大宽度。过大将被等比缩小   | N  | 不限        |                |
| maxHeight        | int        | 图片的最大高度。过大将被等比缩小   | N  | 不限        |                |
| allowEdit        | bool       | 是否允许编辑(框选)。为true时，拍照时会有一个方形的选框   | N  | false     |                |
| useFrontCamera   | bool       | 为false表示后置摄像头，为true表示前置摄像头，如果设备不支持前置摄像头，则为后置摄像头。   | N  | false     | 9.0 (only iOS) |
| multimediaConfig | dictionary | 当multimediaConfig结点存在时，选择 APMultimedia 作为上传存储位置，忽略upload和uploadTo结点<br>可配置的键值：<br>business 该项业务存储标识，默认为“NebulaBiz”<br>compress 压缩等级，默认为4。 0-低质量，1-中质量，2-高质量，3-不压缩，4-根据网络情况自动选择 | N  |           | 9.9            |
| storeOriginal    | bool       | 图片拾取上传时，可以选择是否返回“未经quality/maxWidth/maxHeight/compress等入参导致的有损变换”的”用户选择（或编辑）后的原始图片”的物理暂存路径。  | N  | false     | 9.9            |

3.2. 出参

回调函数带入的参数result: {error, dataURL, fileURL, originalFileURL, useFrontCamera, scene, multimediaID}

| 名称              | 类型     | 描述   | 版本  |
|-----------------|--------|--|-----|
| error           | int    | 错误码（10：用户取消；11：操作失败（权限不够））                           |     |
| dataURL         | string | base64编码的图片数据  |     |
| fileURL         | string | 图片文件URL  |     |
| originalFileURL | string | 相册或拍摄（编辑后）的原图的物理暂存路径                                 |     |
| useFrontCamera  | bool   | 标识拍照时用户是否选择了前置摄像头，为true表示前置摄像头，为false表示后置摄像头         | 9.6 |
| scene           | string | 标识用户获取照片的渠道，值为camera或assets，camera代表拍照，assets代表从相册选择 | 9.6 |
| multimediaID    | string | 标识上传图片在APMultimedia中的identifier                      | 9.9 |

# selectAddress

获取用户收货地址

## 1. 使用方法

```
am.selectAddress(function (data) {  
    alert(JSON.stringify(data));  
});
```

## 2. 代码演示

### 2.1. 基本功能



<h1>点击获取收货地址</h1>

<a href="#" class="btn btn-default read">获取地址</a>

```
<!--必须首先引入该js库-->
<script src="https://publicexpd.alipay.com/deliveraddress/selectAddress.js"></script>
<script>
document.querySelector('.read').addEventListener('click', function(){
    am.selectAddress(function (data) {
        alert(JSON.stringify(data));
    });
});
</script>
```

3. API

```
am.selectAddress(fn);
```

3.1. 入参

| 名称 | 类型       | 描述   | 必选 | 默认值 | 版本 |
|----|----------|------|----|-----|----|
| fn | function | 回调函数 | N  |     |    |

3.2. 出参

回调函数带入的参数result: {address, city, area, addressId, addressCode, post, mobilePhone, prov, fullname}

| 名称          | 类型     | 描述        |
|-------------|--------|-----------|
| address     | string | 街道地址      |
| city        | string | 城市        |
| area        | string | 地区（比如西湖区） |
| addressCode | string | 地址码       |
| addressId   | string | 地址id      |
| mobilePhone | string | 手机电话      |
|             |        |           |

|          |        |      |
|----------|--------|------|
| prov     | string | 省份   |
| fullname | string | 用户全名 |

Since 9.6

# startShare

唤起分享面板进行分享。

分享的默认内容为服务端根据一定的规则自动抓取的当前页面内容而生成。

可通过JSAPI入参传入onlySelectChannel的方式进行分享内容自定义，或通过meta标签进行自定义。

## 1. 注意:

- 无法直接控制显示的分享渠道列表，只能通过bizType入参来控制分享渠道列表的内容和排序。可向分享组件的@亦德 申请bizType。
- 唤起的分享面板第一栏才是分享渠道列表，第二栏为工具栏。如需自定义第二栏工具栏的内容，需要调用 setToolbarMenu 接口进行自定义。

## 2. 分享渠道名称列表

目前已支持的分享渠道名分别为：

```
"ALPContact" : 支付宝好友
"ALPTimeLine" : 支付宝我的生活
"ALPCommunity" : 支付宝生活圈
"Weibo" : 新浪微博
"DingTalkSession" : 钉钉
"SMS" : 短信
"Weixin" : 微信
"WeixinTimeLine" : 微信朋友圈
"QQ" : QQ
"QQZone" : QQ空间
"OpenInSafari" : 在浏览器中打开（不支持onlySelectChannel）
"Favorite" : 收藏（不支持onlySelectChannel）
```

## 3. 使用方法

```
AlipayJSBridge.call("startShare",{
  'bizType':'testShareBizType', //业务标识, 为空时将会显示默认 的分享渠道列表。如需自定义分享渠道, 需要向分享组件的@亦德 申请业务标识
  'onlySelectChannel':["ALPContact","Weixin"] // 当用户选择该数组内指定的分享渠道时, 仅返回渠道名, 而不是真正开始分享
},function(data){
  console.log(data)
});
```

## 4. 代码演示

### 4.1. 从meta读取分享信息

```
<!--
// 钱包9.0开始支持 -->
<meta name="Alipay:title" content="我是标题"/>
<!--
自定义图片
// 钱包8.6开始支持-->
<meta name="Alipay:imgUrl" content="http://xxx/url/xxx.png"/>
<!--
自定义摘要
// 钱包8.6开始支持-->
<meta name="Alipay:desc" content="我是摘要信息"/>
<!--
自定义链接
// 钱包10.0.0开始支持-->
<meta name="Alipay:link" content="http://www.alipay.com"/>

<h1>请点击右上角。。。, 然后分享到相关渠道</h1>
```

这种方法无法根据分享的渠道动态更换内容

### 4.2. 根据渠道进行自定义分享

<h1>请点击按钮进行测试不同渠道分享</h1>

<a href="javascript:void(0)" class="btn btn-default read">开始分享</a>

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function() {
  document.querySelector('.read').addEventListener('click', function() {
    AlipayJSBridge.call("startShare", {
      // 当用户选择该数组内指定的分享渠道时, 仅返回渠道名, 而不是真正开始自动分享
      'onlySelectChannel': ["Weibo", "ALPContact", "ALPTimeLine", "SMS", "Weixin", "WeixinTimeLine", "QQ", "QQZone","DingTalkSession", "OpenInSafari", "Favorite"]
    }, function(data) {
      console.log("当前用户点击的分享渠道名为: "+data.channelName);
      // 通过onlySelectChannel屏蔽掉自动分享功能后, 自行调用shareToChannel接口进行单独分享
      AlipayJSBridge.call("shareToChannel", {
        name: data.channelName,
        param: {
          contentType: 'url',      // 选填, 目前支持支持"auto",text","image","url" 格式 (android分享组件不支持auto)
          title: '自定义分享的标题',
          content: '自定义分享的内容',
          iconUrl: 'https://zos.alipayobjects.com/rmsportal/CKajyImsyyUtTEkzEqcM.png',
          imageUrl: 'https://zos.alipayobjects.com/rmsportal/CKajyImsyyUtTEkzEqcM.png',
          captureScreen: false, // 是否分享当前页面的截图
          url: 'https://m.taobao.com',
          otherParams: {} // 特定分享渠道的扩展参数
        }
      }, function(result) {
        console.log(result)
      });
    });
  });
});
</script>
```

## 5. API

```
AlipayJSBridge.call('startShare', {
  bizType, onlySelectChannel
}, fn);
```

### 5.1. 入参

| 名称                | 类型       | 描述   | 必选 | 默认值       | 版本 |
|-------------------|----------|--|----|-----------|----|
| bizType           | string   | 业务标识，为空时将会显示默认的股份渠道列表。如需自定义分享渠道，需要向分享组件的@亦德 申请业务标识 | N  | 200000067 |    |
| onlySelectChannel | array    | 当用户选择该数组内指定的分享渠道时，仅返回渠道名，而不是真正开始分享                 | N  |           |    |
| fn                | function | 选择渠道后的回调   | N  |           |    |

### 5.2. 出参

|   |
|---|
| 回调函数带入的参数result: {channelName, shareResult} |
|---|

| 名称          | 类型     | 描述        | 版本  |
|-------------|--------|-----------|-----|
| channelName | string | 用户选择的渠道名称 | 9.6 |
| shareResult | bool   | 分享的结果     | 9.6 |

## 6. 参考使用

- 支付宝分享接口开发实践 (../fe/share.html)

Since 8.0

开发者在h5页面唤起支付请使用开放平台 alipay.trade.create (https://doc.open.alipay.com/docs/api.htm?spm=a219a.7629065.0.0.yjpXrR&apild=1046&docType=4)  
接口生成交易号

## 1. 使用方法

```
// 调用此jsapi的时候，不要在当前页面有监听resume事件并在resume事件中强制刷新页面。  
// 调用方式1：通过传入交易号唤起快捷调用方式  
AlipayJSBridge.call("tradePay",{  
    tradeNO: "201209071234123221"  
}, function(result){  
});  
// 调用方式2：通过传入传入交易的字符串唤起快捷调用方式，只需要这一个参数  
AlipayJSBridge.call("tradePay",{  
    orderStr: "xxxxxxxxxxxxxxxxxxxxxxxxxx"  
}, function(result){  
});
```

## 2. 代码演示

### 2.1. 交易号唤起收银台

```
<h1>点击以下按钮唤起收银台支付</h1>  
<a href="javascript:void(0)" class="tradeno">交易号唤起</a>  
<script>  
function ready(callback) {  
    // 如果jsbridge已经注入则直接调用  
    if (window.AlipayJSBridge) {  
        callback && callback();  
    } else {  
        // 如果没有注入则监听注入的事件  
        document.addEventListener('AlipayJSBridgeReady', callback, false);  
    }  
}  
ready(function(){  
    document.querySelector('.tradeno').addEventListener('click', function(){  
        AlipayJSBridge.call("tradePay",{  
            tradeNO: "201209071234123221"  
        }, function(result){  
            alert(JSON.stringify(result));  
        });  
    });  
});  
</script>
```

### 2.2. 唤起支付密码校验（不发起支付，只做身份确认）

```
<h1>点击下面按钮唤起密码校验</h1>
<a href="javascript:void(0)" class="tradeStr">唤起密码校验</a>
<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('.tradeStr').addEventListener('click', function(){
        AlipayJSBridge.call("tradePay",{
            orderStr: "uuid=95a76812-0b4c-46a6-b4ae-cc48c97b5fc5"+ // uuid每次唯一
            "&extern_token=GZ00UIvFe6xfGj9asLXhWkUeUJoXyGmobilegwGZ00"+ // extern_token登入sessionId
            "&app_name=alipay"+
            "&biz_type=openservice"+
            "&apiname=com.alipay.paypwd.validate"+
            "&apiservice=com.alipay.loan.agreement.sign"+
            "&tip=需要在密码框上显示的文案"
        }, function(result){
            alert(JSON.stringify(result));
        });
    });
});
</script>
```

2.3. 唤起支付渠道选择列表

```
<h1>点击下面按钮唤起支付渠道选择</h1>
<a href="javascript:void(0)" class="tradeStr">支付渠道</a>
<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('.tradeStr').addEventListener('click', function(){
        AlipayJSBridge.call('getUserInfo', function(info){
            var userId= info.userId;

            AlipayJSBridge.call("tradePay",{
                orderStr: "uuid=95a76812-0b4c-46a6-b4ae-cc48c97b5fc5"+ // uuid每次唯一
                "&extern_token=GZ00cSiae3xXJ9sXjuV2NhnyfW00Pb98mobilegwGZ00"+// extern_token登入sessionId
                "&app_name=alipay"+
                "&biz_identity=userassets10002"+
                "&trade_from=ALIPAY"+
                "&biz_type=switch_channel"
            }, function(result){
                alert(JSON.stringify(result));
            });
        });
    });
});
</script>
```

## 2.4. 唤起添加银行卡



```
<h1>点击下面按钮唤起添加银行卡</h1>
<a href="javascript:void(0)" class="tradeStr">添加银行卡</a>
<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('.tradeStr').addEventListener('click', function(){
        AlipayJSBridge.call("tradePay",{
            orderStr: "uuid=95a76812-0b4c-46a6-b4ae-cc48c97b5fc5"+ // uuid每次唯一
            "&extern_token=GZ00UIvFe6xfGj9asLXhWkUeUJoXyGmobilegwGZ00"+ // extern_token登入sessionId
            "&app_name=alipay"+
            "&biz_type=new_card"+
            "&trade_from=ALIPAY"+
            "&forbid_channel=credit"
        }, function(result){
            alert(JSON.stringify(result));
        });
    });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('tradePay', {
    tradeNO, partnerID, bizType, bizSubType, displayPayResult, bizContext, orderStr
}, fn);
```

#### 3.1. 入参

| 名称         | 类型     | 描述           | 必选 | 默认值     | 版本  |
|------------|--------|--------------|----|---------|-----|
| tradeNO    | string | 交易号。多个用”；”分隔 | N  |         | 8.0 |
| partnerID  | string | 商户id         | N  | “”      | 8.0 |
| bizType    | string | 交易类型         | N  | ‘trade’ | 8.0 |
| bizSubType | string | 交易子类型        | N  | “”      | 8.0 |
|            |        |              |    |         |     |

|                  |          |                    |   |      |     |
|------------------|----------|--------------------|---|------|-----|
| displayPayResult | bool     | 是否显示支付结果页          | N | true | 8.0 |
| bizContext       | string   | 支付额外的参数，格式为JSON字符串 | N | “    | 8.2 |
| orderStr         | string   | 完整的一个支付字符串         | N | “    | 8.2 |
| fn               | function | 收营台关闭之后的结果回调       | N |      |     |

### 3.2. 出参

|  |
|--|
| 回调函数带入的参数result: {resultCode, callbackUrl, memo, result} |
|--|

| 名称          | 类型     | 描述  | 版本                   |
|-------------|--------|---|----------------------|
| resultCode  | string | 支付结果<br>‘9000’：订单支付成功；<br>‘8000’：正在处理中；<br>‘4000’：订单支付失败；<br>‘6001’：用户中途取消；<br>‘6002’：网络连接出错<br>‘99’：用户点击忘记密码快捷界面退出(only iOS since 9.5) | 8.0                  |
| callbackUrl | bool   | 交易成功后应跳转到的url；一般为空，除非交易有特殊配置  | 8.0                  |
| memo        | bool   | 收银台服务端返回的memo字符串  | iOS 8.0, android 8.4 |
| result      | bool   | 收银台服务端返回的result字符串  | iOS 8.0, android 8.4 |

### 3.3. orderStr各个子参数

| 名称           | 类型     | 描述  |
|--------------|--------|---|
| uuid         | string | 随机字符串，可自行生成或服务端接口                                 |
| extern_token | string | 钱包信任登录token，即为sessionId，通过服务端接口获取                 |
| app_name     | string | 调用方app标识，alipay代表支付宝钱包                            |
| biz_type     | string | 调用业务类型，openservice代表开放基础服务                        |
| apiname      | string | 调用开放服务api名称，com.alipay.paypwd.validate代表独立密码校验服务  |
| apiservice   | string | 调用开放服务api业务方，com.alipay.loan.agreement.sign代表小贷签约 |
| biz_identity | string | 某个业务类型对应的具体业务参数                                   |
|              |        |   |

|         |        |                |
|---------|--------|----------------|
| user_id | string | 某些业务需要的用户id    |
| tip     | string | 需要在密码框上展示的提示文案 |

Since 8.2

# vibrate

调用震动

## 1. 使用方法

```
AlipayJSBridge.call('vibrate');
```

## 2. 代码演示

### 2.1. 基本使用

```
<h1>点击按钮手机会震动</h1>

<a href="#" class="btn btn-default read">手机震动</a>

<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.read').addEventListener('click', function(){
    AlipayJSBridge.call('vibrate');
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('vibrate');
```

#### 3.1. 入参

无参数

Since 8.2

## watchShake

摇一摇功能

每次调用API，在摇一摇手机后触发回调事件，然后需要再次调用这个API。

### 1. 使用方法

```
AlipayJSBridge.call("watchShake",function(data){  
    alert(JSON.stringify(data));  
});
```

### 2. 代码演示

#### 2.1. 摇一摇基本功能

<h1>请点击按钮注册摇一摇事件，然后摇动手机</h1>  
<p>这个API调用后，摇一摇才会被感知。同时一旦摇一摇完成，这个事件就结束了，需要再次调用API</p>

<a href="#" class="btn btn-default read">注册摇一摇</a>

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.read').addEventListener('click', function(){
    AlipayJSBridge.call("watchShake",function(data){
      alert(JSON.stringify(data));
    });
  });
});
</script>
```

### 3. API

AlipayJSBridge.call('watchShake', fn);

#### 3.1. 出参

回调函数带入的参数result: {success}

| 名称      | 类型   | 描述      | 版本  |
|---------|------|---------|-----|
| success | bool | 是否摇一摇成功 | 8.6 |

# actionSheet

提供一种供选择选项的列表， 它会dock在屏幕的下边沿

## 1. 使用方法

```
AlipayJSBridge.call('actionSheet',{
  'title': '标题',
  'btns': ['第一个按钮', '第二个按钮', '第三个按钮'],
  'cancelBtn': '取消',
  'destructiveBtnIndex': 2
}, function(data) {
  switch (data.index) {    // index标示用户点击的按钮，在actionSheet中的位置，从0开始
    case 0:
      alert('第一个按钮');
      break;
    case 1:
      alert('第二个按钮');
      break;
    case 2:
      alert('第三个按钮');
      break;
    case 3:
      alert('取消按钮');
      break;
  }
});
```

## 2. 代码演示

### 2.1. 基本功能

```
<h1>点击按钮调出actionSheet</h1>
<a href="javascript:void(0)" class="actionSheet">打开actionSheet</a>
```

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.actionSheet').addEventListener('click', function(){
    AlipayJSBridge.call('actionSheet',{
      'title': '标题',
      'btns': ['第一个按钮', '第二个按钮', '第三个按钮'],
      'cancelBtn': '取消',
      'destructiveBtnIndex': 2
    }, function(data) {
      switch (data.index) {    // index标示用户点击的按钮, 在actionSheet中的位置, 从0开始
        case 0:
          alert('第一个按钮');
          break;
        case 1:
          alert('第二个按钮');
          break;
        case 2:
          alert('第三个按钮');
          break;
        case 3:
          alert('取消按钮');
          break;
      }
    });
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('actionSheet',{
  title, btns, cancelBtn, destructiveBtnIndex
}, fn)
```

### 3.1. 入参

| 名称                  | 类型       | 描述  | 必选 | 默认值 | 版本 |
|---------------------|----------|---|----|-----|----|
| title               | string   | 标题  | N  |     |    |
| btns                | array    | 一组按钮，item类型是string                              | Y  |     |    |
| cancelBtn           | string   | 设置取消按钮及文字                                       | N  |     |    |
| destructiveBtnIndex | int      | （IOS特殊处理）指定按钮的索引号，从0开始，使用场景：需要删除或清除数据等类似场景，默认红色 | N  |     |    |
| fn                  | function | 不是API调用后被回调，而是用户选择选项之后的回调函数                     | N  |     |    |

### 3.2. 出参

|                                      |
|--------------------------------------|
| 格式是 {data: {index: 0}}               |
| 这个index是用户点击的按钮在actionSheet中的位置，从0开始 |

Since 8.1

# alert

alert对话框的native实现

## 1. 使用方法



```
AlipayJSBridge.call('alert', {
  title: '亲',
  message: '你好',
  button: '确定'
}, function(e){
  alert('回调结果: ' + JSON.stringify(e))
});
```

## 2. 代码演示

### 2.1. alert和confirm

```
<h1>点击以下按钮看不同效果</h1>
<a href="javascript:void(0)" class="alert">点击Alert</a>
<a href="javascript:void(0)" class="confirm">点击Confirm</a>
```

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.alert').addEventListener('click', function(){
    AlipayJSBridge.call('alert', {
      title: '亲',
      message: '你好',
      button: '确定'
    }, function(e){
      alert('回调结果: ' + JSON.stringify(e))
    });
  });

  document.querySelector('.confirm').addEventListener('click', function(){
    AlipayJSBridge.call('confirm', {
      title: '亲',
      message: '确定要退出吗? ',
      okButton: '是',
      cancelButton: '否'
    }, function(e){
      alert('回调结果: ' + JSON.stringify(e))
    });
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('alert',{
  title, message, button
}, fn)
```

#### 3.1. 入参

| 名称      | 类型       | 描述                 | 必选 | 默认值  | 版本 |
|---------|----------|--------------------|----|------|----|
| title   | string   | alert框标题           | N  | ‘’   |    |
| message | string   | alert框文本           | N  |      |    |
| button  | string   | 按钮文字               | N  | ‘确定’ |    |
| fn      | function | 回调函数，当点击button后被调用 | N  |      |    |

## 4. 使用注意

- alert 这个api不像window.alert，它不是阻塞式的，也就是说先后弹2个的话，最后看到的是后弹的那个。

Since 8.1

# confirm

confirm确认框的native实现

## 1. 使用方法

```
AlipayJSBridge.call('confirm', {
  title: '亲',
  message: '确定要退出吗? ',
  okButton: '是',
  cancelButton: '否'
}, function(e){
  alert('回调结果: ' + JSON.stringify(e))
});
```

## 2. 代码演示

### 2.1. alert和confirm

```
<h1>点击以下按钮看不同效果</h1>
<a href="javascript:void(0)" class="alert">点击Alert</a>
<a href="javascript:void(0)" class="confirm">点击Confirm</a>
```

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.alert').addEventListener('click', function(){
    AlipayJSBridge.call('alert', {
      title: '亲',
      message: '你好',
      button: '确定'
    }, function(e){
      alert('回调结果: ' + JSON.stringify(e))
    });
  });

  document.querySelector('.confirm').addEventListener('click', function(){
    AlipayJSBridge.call('confirm', {
      title: '亲',
      message: '确定要退出吗? ',
      okButton: '是',
      cancelButton: '否'
    }, function(e){
      alert('回调结果: ' + JSON.stringify(e))
    });
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('alert',{
  title, message, okButton, cancelButton
}, fn)
```

#### 3.1. 入参

| 名称           | 类型       | 描述                 | 必选 | 默认值  | 版本 |
|--------------|----------|--------------------|----|------|----|
| title        | string   | alert框标题           | N  | ‘’   |    |
| message      | string   | alert框文本           | N  |      |    |
| okButton     | string   | 确定按钮文字             | N  | ‘确定’ |    |
| cancelButton | string   | 取消按钮文字             | N  | ‘取消’ |    |
| fn           | function | 回调函数，当点击button后被调用 | N  |      |    |

## 4. 使用注意

- confirm 这个api和 alert 一样，它也不是阻塞式的，也就是说先后弹2个的话，最后看到的是后弹的那个。

Since 8.6

# 自定义键盘

自定义键盘使用：Android和ios方案不一样，Android支持直接通过原生的input输入框唤起，ios通过jsapi唤起

### 0.1. iOS

- 前端DOM结构

```
< input type="text" class="h5numInput" kb-type="0" data-paste="false" maxlength="8" data-placeholder="这里调用数字键盘" />
- type: 支持 type="password"
- class="h5numInput" 标识页面初始化加载就调用自定义键盘；
- kb-type: （9.9.1之后开始支持别名data-kb-type）需要调用的键盘类型，`0`为数字键盘，`1`为身份证键盘，`2`为不带小数点的键盘，不写该属性默认为数字键盘；键盘类型随时可扩展；
- data-paste: 若为 false 则不能进行粘贴，默认为 true；
- data-placeholder: 需要显示的提示信息，不能只写placeholder；
- maxlength: 最大长度
- 居左、中、右对齐：添加 css的 text-align, 支持 left, center, right, justify, 默认为 left
```

- ios数字键盘相关 API（只有ios有，android不支持）

numInput：用于手动调用自定义键盘

```
AlipayJSBridge.call('numInput');
```

inputFocus: 用于当前端校验某input 有问题, focus 到 input 输入框

```
AlipayJSBridge.call('inputFocus', {  
  index: index //用于 focus 的某 input, 从0开始 (带有class含h5numInput的输入框)  
});
```

inputBackFill: 用于当前端去除无用值, 返回正确的值给 native (since 9.0)

```
AlipayJSBridge.call('inputBackFill', {  
  index: index, //用于回填的某 input, 从0开始 (带有class含h5numInput的输入框)  
  value: 111  
});
```

numInputReset: 用于重新渲染某输入框的位置 (since 9.0)

```
AlipayJSBridge.call('numInputReset', {  
  index: index //用于重新渲染 input 位置, 从0开始 (带有class含h5numInput的输入框)  
});
```

inputBlur: 手动blur (since 9.5)

```
AlipayJSBridge.call('inputBlur', function() {  
  // do something  
});
```

## 0.1. 回调事件

```
document.addEventListener('inputEvent', function(event) {
  console.log("index:" + event.data.index);
  console.log("number:" + event.data.value);
  console.log("eventType:" + event.data.eventType);

  if(event.data.eventType == "blur") {
    // do blur
  }
  else if(event.data.eventType == "focus") {
    // do focus
  }
  else if(event.data.eventType == "keyup") {
    // do keyup
  }
  else if(event.data.eventType == "keydown") {
    // do keydown
  }
  else if(event.data.eventType == "change") {
    // do change
  }
}, false);
```

## 1. 使用方法

## 2. 代码演示

### 2.1. 基本功能

```
<style>
* {
  padding: 0;
  margin: 0;
}
input {
  display: block;
  padding: 8px;
  box-sizing: border-box;
  -webkit-box-sizing: border-box;
  margin: 10px auto;
}
#J_inputLog {
  line-height: 1.2;
  font-size: 12px;
}
</style>
```

```
<br>
<input
  type="text"
  class="h5numInput"
  data-kb-type="0"
  kb-type="0"
  data-paste="false"
  maxlength="6"
  data-placeholder="iOS kb-type=0"
  id="J_input1"
/>
```

```
<br>
<input
  type="text"
  class="h5numInput"
  data-kb-type="1"
  kb-type="1"
  data-paste="false"
  maxlength="6"
  data-placeholder="iOS kb-type=1"
  id="J_input2"
/>
```

```
<br>
<input
  type="text"
  class="h5numInput"
  data-kb-type="2"
  kb-type="2"
  data-paste="false"
  maxlength="6"
  data-placeholder="iOS kb-type=2"
  id="J_input3"
/>
```

```
<br>
<input
  type="number"
  placeholder="system/Android"
  id="J_input4"
/>
```

<h5>键盘事件日志</h5>

<div id="J\_inputLog"></div>

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
```



```

    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function() {
    function doLog(logStr) {
        document.getElementById('J_inputLog').innerHTML += logStr + '<br>';
    }

    document.addEventListener('inputEvent', function(event) {
        doLog("index:" + event.data.index);
        doLog("number:" + event.data.value);
        doLog("eventType:" + event.data.eventType);

        console.log(event);

        if(event.data.eventType == "blur") {
            // do blur
        }
        else if(event.data.eventType == "focus") {
            // do focus
        }
        else if(event.data.eventType == "keyup") {
            // do keyup
        }
        else if(event.data.eventType == "keydown") {
            // do keydown
        }
        else if(event.data.eventType == "change") {
            // do change
        }
    }, false);
});
</script>

```

## 2.2. Android

安卓自定义键盘直接识别type='number'， 'tel'或者其他以正则表达式约束的input输入框， 'tel'支持input的其他扩展项，如maxLength等。如果当前页面存在其他输入框，可在数字类型输入框的onfocus调用'hideKeyboard'接口，以隐藏显示的系统键盘之后再唤起自定义键盘。 < input type="number" data-placeholder="这里调用数字键盘" />

hideAllKeyboard：隐藏所有键盘（包括自定义键盘）（since 10.0.15）

```
AlipayJSBridge.call('hideAllKeyboard');
```

### 2.3. Android数字键盘demo

```
<h1>Android自定义数字键盘</h1>
<input type="number" data-placeholder="这里调用数字键盘"/>
```

Since 8.0

# hideLoading

隐藏全局大菊花

## 1. 使用方法

```
AlipayJSBridge.call('hideLoading');
```

## 2. 代码演示

### 2.1. 显示/隐藏全局loading框

<h1>点击以下按钮看不同效果</h1>

<p>注意安卓下显示loading后，会覆盖整个界面，所以请使用系统回退键关闭loading</p>

<a href="javascript:void(0)" class="show">显示loading</a>

<a href="javascript:void(0)" class="hide">隐藏loading</a>

<a href="javascript:void(0)" class="delay">延迟2秒显示loading</a>

<a href="javascript:void(0)" class="notext">无文字菊花</a>

<script>

function ready(callback) {

  // 如果jsbridge已经注入则直接调用

  if (window.AlipayJSBridge) {

    callback && callback();

  } else {

    // 如果没有注入则监听注入的事件

    document.addEventListener('AlipayJSBridgeReady', callback, false);

  }

}

ready(function(){

  document.querySelector('.show').addEventListener('click', function(){

    AlipayJSBridge.call('showLoading', {

      text: '加载中',

    });

});

  document.querySelector('.hide').addEventListener('click', function(){

    AlipayJSBridge.call('hideLoading');

});

  document.querySelector('.delay').addEventListener('click', function(){

    AlipayJSBridge.call('showLoading', {

      text: '加载中',

      delay: 2000,

    });

});

  document.querySelector('.notext').addEventListener('click', function(){

    AlipayJSBridge.call('showLoading', {

    });

});

});

</script>

# hideOptionsMenu

隐藏标题栏右边的按钮属性

## 1. 使用方法

```
AlipayJSBridge.call('hideOptionsMenu');
```

## 2. 代码演示

请参考setOptionsMenu (./set-option-menu.html)

Since 8.6

# hideTitleLoading

隐藏标题栏loading

## 1. 使用方法

```
AlipayJSBridge.call('hideTitleLoading');
```

## 2. 代码演示

### 2.1. 显示/隐藏全局loading框

```
<h1>点击下面按钮查看效果</h1>
<a href="javascript:void(0)" class="show">显示loading</a>
<a href="javascript:void(0)" class="hide">隐藏loading</a>

<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.show').addEventListener('click', function(){
    AlipayJSBridge.call('showTitleLoading');
  });

  document.querySelector('.hide').addEventListener('click', function(){
    AlipayJSBridge.call('hideTitleLoading');
  });
});
</script>
```

# prompt

prompt输入框弹窗的native实现

## 1. 使用方法

```
AlipayJSBridge.call('prompt', {
  title: '亲',
  message: '确定要这么干吗? ',
  placeholder:'提示文案',
  okButton: '是',
  cancelButton: '否'
}, function(result){
  // result.ok      bool型  用户是否点击确定
  // result.inputValue  用户输入的内容
  console.log(result.ok,result.inputValue);
});
```

## 2. 代码演示

### 2.1. prompt

```
<h1>点击以下按钮看效果</h1>
<a href="javascript:void(0)" class="prompt">点击prompt</a>
<p class="result"></p>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.prompt').addEventListener('click', function(){
    AlipayJSBridge.call('prompt', {
      title: '亲',
      message: '确定要这么干吧',
      okButton: '确定',
      cancelButton: '算了',
      placeholder:'placeholder'
    }, function (result) {
      // result.ok      bool型   用户是否点击确定
      // result.inputValue   用户输入的内容
      if (result.ok) {
        document.querySelector('.result').textContent = result.inputValue;
      }
    });
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('prompt',{
  title, message, okButton, cancelButton,placeholder
}, fn)
```

#### 3.1. 入参

| 名称      | 类型     | 描述        | 必选 | 默认值 | 版本 |
|---------|--------|-----------|----|-----|----|
| title   | string | prompt框标题 | N  | “   |    |
| message | string | prompt框文本 | N  |     |    |

|              |          |                    |   |      |  |
|--------------|----------|--------------------|---|------|--|
| okButton     | string   | 确定按钮文字             | N | ‘确定’ |  |
| cancelButton | string   | 取消按钮文字             | N | ‘取消’ |  |
| placeholder  | string   | 输入框内的提示文案          | N | ‘’   |  |
| fn           | function | 回调函数，当点击button后被调用 | N |      |  |

## 4. 使用注意

- prompt 这个api和 alert 一样，它也不是阻塞式的，也就是说先后弹2个的话，最后看到的是后弹的那个。

# setBarBottomLineColor

可通过该接口自定义导航栏底部细线的颜色，当颜色与导航栏相同时可达到隐藏的效果

## 1. 使用方法

```
AlipayJSBridge.call("setBarBottomLineColor", {
  color: 16711688
});
```

## 2. 代码演示



```
<div style="padding-top:80px;">
  <a href="javascript:void(0)" class="title">设置导航栏底部细线颜色</a><br>
</div>

<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.title').addEventListener('click', function(){
    AlipayJSBridge.call("setBarBottomLineColor", {
      color: 16711688
    });
  },false);
},false);
</script>
```

### 3. API

```
AlipayJSBridge.call('setBarBottomLineColor',{
  color
}, fn)
```

#### 3.1. 入参

| 名称    | 类型  | 描述     | 必选 | 默认值 | 版本 |
|-------|-----|--------|----|-----|----|
| color | int | 色值，十进制 | Y  |     |    |

Since 8.0

## setOptionsMenu

设置标题栏右边的按钮属性。

该接口仅负责设置，需要额外调用showOptionsMenu保证该按钮的显示。

注意：由于苹果的ATS限制，icon URL必须为https链接或base64，http链接会被忽略

## 1. 使用方法

```
AlipayJSBridge.call('setOptionsMenu', {
  title : '按钮', // 与icon、icontype三选一
  redDot : '-1', // -1表示不显示, 0表示显示红点, 1-99表示在红点上显示的数字
  color : '#ff00ff00', // 必须以#开始ARGB颜色值
});
```

## 2. 代码演示

### 2.1. 设置各种类型的右按钮

```
<h1>点击以下按钮看不同效果</h1>

<a href="javascript:void(0)" class="button">单按钮</a>
<a href="javascript:void(0)" class="icon">单图标</a>
<a href="javascript:void(0)" class="menu">下拉菜单(9.9.3)</a>
<a href="javascript:void(0)" class="reset">重置</a>
<a href="javascript:void(0)" class="hide">隐藏</a>
<a href="javascript:void(0)" class="show">显示</a>

<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(e){
  document.querySelector('.button').addEventListener('click', function(){
    AlipayJSBridge.call('setOptionsMenu', {
      title : '按钮',
      redDot : '5', // -1表示不显示, 0表示显示红点, 1-99表示在红点上显示的数字
      color : '#ff00ff00', // 必须以#开始ARGB颜色值
    });
    AlipayJSBridge.call('showOptionsMenu');
```

```
});

document.querySelector('.icon').addEventListener('click', function(){
    AlipayJSBridge.call('setOptionsMenu', {
        icon : 'http://pic.alipayobjects.com/e/201212/1nt0VeWwtg.png',
        redDot : '-1', // -1表示不显示, 0表示显示红点, 1-99表示在红点上显示的数字
    });
    AlipayJSBridge.call('showOptionsMenu');
});

document.querySelector('.menu').addEventListener('click', function(){
    AlipayJSBridge.call('setOptionsMenu', {
        // 显示的时候是从后往前显示的
        menus: [{
            icontype: 'scan',
            redDot: '-1', // -1表示不显示, 0表示显示红点, 1-99表示在红点上显示的数字
        }, {
            icontype: 'user',
            redDot: '-1', // -1表示不显示, 0表示显示红点, 1-99表示在红点上显示的数字
        }],
        override: true // 在需要设置多个option的情况下, 是否保留默认的optionMenu
    });

    // 必须强制调用一把刷新界面
    AlipayJSBridge.call('showOptionsMenu');
});

document.querySelector('.reset').addEventListener('click', function(){
    AlipayJSBridge.call('setOptionsMenu', {
        reset: true,
    });
    AlipayJSBridge.call('showOptionsMenu');
});

document.querySelector('.show').addEventListener('click', function(){
    AlipayJSBridge.call('showOptionsMenu');
});

document.querySelector('.hide').addEventListener('click', function(){
    AlipayJSBridge.call('hideOptionsMenu');
});

document.addEventListener('optionMenu', function (e) {
    alert(JSON.stringify(e));
}, false);
});
</script>
```

### 3. API

要注意的是有几个属性是优先级的：reset > title > icontype > icon， 这4个属性只需要有一个即可。

```
AlipayJSBridge.call('setTitle',{
  title, icon, redDot, reset, color, override, menus, icontype
})
```

#### 3.1. 入参

| 名称             | 类型     | 描述  | 必选 | 默认值       | 版本    |
|----------------|--------|---|----|-----------|-------|
| title          | string | 右按钮文字   | Y  |           |       |
| icon           | string | 右按钮图标url、base64（since 9.0）<br>8.3及以前：iOS 40x40（周边不留白），Android 50x50（四边各透明留白5px）<br>8.4及以后：两个平台统一使用40x40（周边不留白）  | Y  |           |       |
| redDot         | string | 红点数值  | N  |           | 8.6   |
| reset          | bool   | 重置到系统默认，当reset=true时，忽略其他参数   | Y  | false     | 8.6   |
| color          | string | 文字颜色值   | N  | #ffffffff | 9.0   |
| override       | bool   | 在需要设置多个option的情况下，是否保留默认的optionMenu   | N  | false     | 9.9   |
| menus          | array  | 设置多个按钮  | N  | []        | 9.9   |
| preventDefault | bool   | 是否阻止默认分享功能（默认是弹分享框）preventDefault=true会阻止默认分享   | N  | []        | 9.9   |
| icontype       | string | 根据图片类型加载容器预置图片,与title、icon三选一。<br>注意：只支持单个optionMenu变色。<br>具体类型包含：user（账户）、filter（筛选器）、search（查找）、add（添加）、settings（设置）、scan（扫一扫）、info（信息）、help（帮助）、locate（定位）、more(更多)、mail(邮箱 10.0.8及以上) | N  |           | 9.9.3 |

### 4. 使用注意

- setOptionsMenu 后如果效果不对，请调用一把 showOptionsMenu

# setTitleColor

可通过启动参数和接口调用， 改变TitleBar的颜色

## 1. 使用方法

```
AlipayJSBridge.call("setTitleColor", {
  color: 16775138,
  reset: false //(可选,默认为false)  是否重置title颜色为默认颜色。
});
```

## 2. 代码演示

```
<div style="padding-top:80px;">
  <a href="javascript:void(0)" class="title">设置标题颜色</a><br>
  <a href="javascript:void(0)" class="reset">重置标题颜色</a><br><br>

  <a href="javascript:void(0)" class="pushWindow">新开查看透明标题重置效果</a><br>
  <a href="javascript:void(0)" class="resetTransparent">重置透明标题颜色</a>
</div>
```

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.title').addEventListener('click', function(){
    AlipayJSBridge.call("setTitleColor", {
      color: 16775138,
      reset: false //(可选,默认为false) 是否重置title颜色为默认颜色。
    });
  });

  document.querySelector('.reset').addEventListener('click', function(){
    AlipayJSBridge.call("setTitleColor",{
      color: 16775138,
      reset: true
    });
  });

  document.querySelector('.pushWindow').addEventListener('click', function(){
    AlipayJSBridge.call("pushWindow", {
      url: './set-title-color-demo.html?__webview_options__=transparentTitle%3Dalways'
    });
  });

  document.querySelector('.resetTransparent').addEventListener('click', function(){
    AlipayJSBridge.call("setTitleColor",{
      color: 16775138,
      resetTransparent: true
    });
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('setTitleColor',{
  color: 16775138,
  reset: false,
  resetTransparent: false
}, fn)
```

#### 3.1. 入参

| 名称               | 类型   | 描述                   | 必选 | 默认值   | 版本 |
|------------------|------|----------------------|----|-------|----|
| color            | int  | 色值，十进制               | Y  |       |    |
| reset            | bool | 是否重置为支付宝默认颜色         | N  | false |    |
| resetTransparent | bool | 如果当前是透明的导航头，重置为默认的白色 | N  | false |    |

Since 8.0

## setTitle

设置页面的标题栏，包括主标题，副标题以及标题菜单项

注意：由于苹果的ATS限制，image URL必须为https链接或base64，http链接会被忽略

#### 1. 使用方法

```
AlipayJSBridge.call('setTitle', {
  title: '标题',
});
```

#### 2. 代码演示

##### 2.1. 设置各种类型的标题栏

```
<h1>点击以下按钮看不同效果</h1>
<a href="javascript:void(0)" class="title">只设置标题</a>
<a href="javascript:void(0)" class="subTitle">标题+副标题</a>
<a href="javascript:void(0)" class="clear">清空标题</a>

<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.title').addEventListener('click', function(){
    AlipayJSBridge.call('setTitle', {
      title: '标题',
    });
  });

  document.querySelector('.subTitle').addEventListener('click', function(){
    AlipayJSBridge.call('setTitle', {
      title: '标题',
      subtitle: '副标题',
    });
  });

  document.querySelector('.clear').addEventListener('click', function(){
    AlipayJSBridge.call('setTitle', {
      title: ' ', // 不是真的清空, 只是设置为空格, 如果注册了title事件, 还是会触发
    });
  });
});
</script>
```

3. API

```
AlipayJSBridge.call('setTitle',{
  title, subtitle, image
}, fn)
```

3.1. 入参

|  |  |  |  |  |  |
|--|--|--|--|--|--|
|  |  |  |  |  |  |
|--|--|--|--|--|--|



| 名称       | 类型     | 描述  | 必选 | 默认值 | 版本    |
|----------|--------|---|----|-----|-------|
| title    | string | 主标题文案   | N  |     |       |
| subtitle | string | 副标题文案   | N  |     |       |
| image    | string | 支持url或者base64，请使用一张3X图，如果设置了image，则前两个参数失效，并且不从webview的回调中读取title | N  |     | 9.9.5 |

Since 8.2

# setToolbarMenu

自定义分享面板第二栏的工具栏内容。

**注意：** 由于苹果的ATS限制，icon URL必须为https链接或base64，http链接会被忽略

## 1. 使用方法

```
AlipayJSBridge.call('setToolbarMenu', {
  menus: [
    {
      name: "字体",
      tag: "H5MenuActionOfFont",
      icon: "https://os.alipayobjects.com/rmsportal/SuHlTsUNSyJvRzn.png"
    },
    {
      name: "菜单项2",
      tag: "cm2"
    },
    {
      name: "菜单项3",
      tag: "cm3"
    }
  ],
  override: false
}, function() {
  // do something
});
```

## 2. 代码演示

## 2.1. 设置工具栏菜单项列表

```
<button id="J_setOverrideValue">override(false)</button><br>
<button id="J_setResetValue">reset(false)</button><br>

<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(e) {
    var options = {
        menus:[
            {
                name: "字体",
                tag: "H5MenuActionOfFont",
                icon: "https://os.alipayobjects.com/rmsportal/SuHlTsUNSyJvRzn.png"
            },
            {
                name: "菜单项2",
                tag: "cm2"
            },
            {
                name: "菜单项3",
                tag: "cm3"
            }
        ],
        override: false,
        reset: false
    };

    document.querySelector('#J_setOverrideValue').addEventListener('click', function(e) {
        options.override = !options.override;
        e.target.innerText = 'override(' + options.override.toString() + ')';
        updateOptions();
    }, false);

    document.querySelector('#J_setResetValue').addEventListener('click', function(e) {
        options.reset = !options.reset;
        e.target.innerText = 'reset(' + options.reset.toString() + ')';
        updateOptions();
    }, false);

    function updateOptions() {
```

```
AlipayJSBridge.call('setToolbarMenu', options, function(result) {
    alert(JSON.stringify(result));
});
}

updateOptions();

document.addEventListener('toolbarMenuClick', function (e) {
    alert(JSON.stringify(e));
}, false);
});
</script>
```

### 3. API

```
AlipayJSBridge.call('setToolbarMenu',{
    menus: {
        name, tag, icon
    }, override, reset
})
```

#### 3.1. 入参

| 名称       | 类型    | 描述   | 必选 | 默认值   | 版本 |
|----------|-------|--|----|-------|----|
| menus    | array | 菜单列表，默认显示于容器默认菜单项的右上角,icon使用容器默认icon（五角星）。<br>name:表示菜单项显示内容，必须控制在4个字符（含）以内，否则容器会截断4个字显示（必须填写）；<br>tag:表示菜单项的唯一标签（必须填写）；<br>icon:可自定义icon，传入图片url（非必填）；<br>当菜单被点击后，传入的值都会被返回；<br>当override为false时，用户自定义菜单个数必须小于5个。 | Y  |       |    |
| override | bool  | 是否覆盖容器自带的菜单列表  | N  | false |    |
| reset    | bool  | 是否重置菜单到系统默认，当reset=true时，忽略其他参数  | N  | false |    |

#### 3.2. 出参

|                           |
|---------------------------|
| 回调函数带入的参数result: {result} |
|---------------------------|

| 名称            | 类型     | 描述                | 版本 |
|---------------|--------|-------------------|----|
| result.status | string | “Success”:添加菜单项成功 |    |

Since 8.0

# showLoading

显示全局大菊花

## 1. 使用方法

```
AlipayJSBridge.call('showLoading', {
  text: '加载中',
});
```

## 2. 代码演示

### 2.1. 显示/隐藏全局loading框

```
<h1>点击以下按钮看不同效果</h1>
<p>注意安卓下显示loading后，会覆盖整个界面，所以请使用系统回退键关闭loading</p>
<a href="javascript:void(0)" class="show">显示loading</a>
<a href="javascript:void(0)" class="hide">隐藏loading</a>
<a href="javascript:void(0)" class="delay">延迟2秒显示loading</a>
<a href="javascript:void(0)" class="notext">无文字菊花</a>
```

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.show').addEventListener('click', function(){
    AlipayJSBridge.call('showLoading', {
      text: '加载中',
    });
  });

  document.querySelector('.hide').addEventListener('click', function(){
    AlipayJSBridge.call('hideLoading');
  });

  document.querySelector('.delay').addEventListener('click', function(){
    AlipayJSBridge.call('showLoading', {
      text: '加载中',
      delay: 2000,
    });
  });

  document.querySelector('.notext').addEventListener('click', function(){
    AlipayJSBridge.call('showLoading', {
    });
  });
});
</script>
```

### 3. API

```
AlipayJSBridge.call('showLoading',{
    text, delay
})
```

### 3.1. 入参

| 名称         | 类型     | 描述   | 必选 | 默认值  | 版本      |
|------------|--------|--|----|------|---------|
| text       | string | 文本内容；若不指定，则显示为中间大菊花；如果指定，显示为小菊花右侧带文字                             | N  | ‘’   |         |
| delay      | int    | 延迟多少毫秒后显示；如果在此时间之前调用了hideLoading，则不会再显示                          | N  | 0    |         |
| autoHide   | bool   | 默认情况下容器会在pageFinish的后会主动隐藏菊花，默认true，传入false，关掉自动隐藏(only android) | N  | true | 10.0.15 |
| cancelable | bool   | 安卓返回键是否消掉菊花，默认物理返回键会消掉菊花(only android)                           | N  | true | 10.0.18 |

## 4. 使用注意

- 注意安卓下显示loading后，会覆盖整个界面，所以请使用系统回退键关闭loading;
- ios下的情况是，当没有设置text值的时候，只有标题栏和工具栏可以被点击， 有文字的时候不能覆盖任何内容。9.9.5以后已修复

Since 8.0

# showOptionsMenu

显示标题栏右边的按钮属性

## 1. 使用方法

```
AlipayJSBridge.call('showOptionsMenu');
```

## 2. 代码演示

请参考setOptionsMenu (./set-option-menu.html)

Since 9.9

# showPopupMenu

右上角显示弹出的下拉菜单,如果设置了icon那么每一个菜单项都要求有icon,要不都没有icon。

特别注意，请右上角为默认三个点的时候不要调用本接口，右上角为自定义图片或者文字的时候可以调用。

注意：tag是必选项 该API在9.9版本之后可以使用

## 1. 使用方法

```
AlipayJSBridge.call('showPopupMenu',{
  menus:[
    {name:"字体", tag:"tag1", redDot:"101"},
    {name:"菜单项2",  tag:"tag1", redDot:"0"},
    {name:"菜单项3",  tag:"tag1"}
  ]
}, function () {

});
```

## 2. 代码演示

### 2.1. 设置弹出菜单栏

```
<h1>请点击右上角弹出的菜单</h1>
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function() {

  AlipayJSBridge.call('setOptionsMenu', {
    title : '点我',
    redDot : '-1'
  });
  AlipayJSBridge.call('showOptionsMenu');

  document.addEventListener('optionMenu', function (e) {
    AlipayJSBridge.call('showPopupMenu', {
      menus:[
        {name:"字体", tag:"tag1",redDot:"101"},
        {name:"菜单项2", tag:"tag2",redDot:"0"},
        {name:"菜单项3", tag:"tag3"},
        {name:"菜单项4", tag:"tag4",redDot:"66"}
      ]
    }, function (e) {
      console.log(e);
    });
  }, false);

  document.addEventListener('popMenuClick', function (e) {
    alert(JSON.stringify(e));
  }, false);
});
</script>
```

### 3. API

```
AlipayJSBridge.call('showPopupMenu',{
  menus: [{name, tag, icon,redDot}]
}, fn)
```

#### 3.1. 入参



| 名称     | 类型       | 描述   | 必选 | 默认值 | 版本 |
|--------|----------|--|----|-----|----|
| menus  | array    | 菜单列表，显示于容器默认菜单项的右上角。<br>name:表示菜单项显示内容，建议在4个字符以内；<br>icon:可自定义icon，传入图片url（非必填）图片大小iOS中需要为42*42大小的图片；<br>当菜单被点击后，点击的的name会返回会通过popMenuClick事件返回； | Y  |     |    |
| icon   | string   | 右按钮图标url   | N  |     |    |
| name   | string   | 菜单显示的内容  | Y  |     |    |
| tag    | string   | tag用于区分选项  | Y  |     |    |
| redDot | string   | 是否显示小红点  | N  |     |    |
| fn     | function | api调用后的回调函数  | N  |     |    |

Since 8.6

# showTitleLoading

在标题栏显示loading

## 1. 使用方法

```
AlipayJSBridge.call('showTitleLoading');
```

## 2. 代码演示

### 2.1. 显示/隐藏全局loading框

```
<h1>点击下面按钮查看效果</h1>
<a href="javascript:void(0)" class="show">显示loading</a>
<a href="javascript:void(0)" class="hide">隐藏loading</a>

<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('.show').addEventListener('click', function(){
        AlipayJSBridge.call('showTitleLoading');
    });

    document.querySelector('.hide').addEventListener('click', function(){
        AlipayJSBridge.call('hideTitleLoading');
    });
});
</script>
```

Since 10.0.18

# showUCFailDialog

uc初始化异常提醒框 showUCFailDialog

## 1. 使用方法

```
AlipayJSBridge.call('showUCFailDialog', {
}, function(e){

});
```

## 2. 代码演示

## 2.1. showUCFailDialog

```
<h1>点击以下按钮看不同效果</h1>
<a href="javascript:void(0)" class="showUCFailDialog">点击showUCFailDialog</a>

<script>
function ready(callback) {
    // 如果jsbridge已经注入则直接调用
    if (window.AlipayJSBridge) {
        callback && callback();
    } else {
        // 如果没有注入则监听注入的事件
        document.addEventListener('AlipayJSBridgeReady', callback, false);
    }
}
ready(function(){
    document.querySelector('.showUCFailDialog').addEventListener('click', function(){
        AlipayJSBridge.call('showUCFailDialog', {

        }, function(e){
            alert('回调结果: ' + JSON.stringify(e))
        });
    });

});
</script>
```

## 3. API

```
AlipayJSBridge.call('showUCFailDialog',{
}, fn)
```

## 4. 使用注意

- 通过ua判断当前webview是不是uc内核：

```
if(ua.indexOf(“UCBS”) !=-1) { “UC 内核”; }
```

附：

U4内核的useragent：

# tabBar

底部tabBar控件

## 1. 使用方法

```
AlipayJSBridge.call('tabBar', {
  action: 'create', // 创建 Tab Bar, 必选
  textColor: 16776960, //ffff00 黄, 必选
  selectedColor:16711680, //ff0000 红, 必选
  selectedIndex:1, // 必选
  items: [ // 所有 TabItems, 必选
    {
      name: 'tab1', // 标题, 必选
      tag: 'tab11', // tab项id, 必选
      icon: 'https://zos.alipayobjects.com/rmsportal/I0wBUWsF0QyNYhhAxDgb.png', // 正常状态图片 URL, 支持 base64 编码 URL, 必选
      activeIcon: 'https://zos.alipayobjects.com/rmsportal/KdyZNLUKgFfAAHreQzsj.png', // 选中状态图片 URL, 支持 base64 编码 URL, 必选
      redDot: -1, // 选填, '0' 表示红点, null、'-1' 表示不显示, 其他值展示出来
    },
    {
      name: 'tab2',
      tag: 'tab22',
      icon: 'https://zos.alipayobjects.com/rmsportal/dMHfRGFxWiRidgHSLLvS.png', // 灰色
      activeIcon: 'https://zos.alipayobjects.com/rmsportal/tNPTHyWzgEQNlZLxHVRl.png', // 选中状态图片 URL, 支持 base64 编码 URL, 必选
      redDot: '99+'
    },
    {
      name: 'tab3',
      tag: 'tab33',
      icon: 'https://zos.alipayobjects.com/rmsportal/neFZyipaNvaZiMgNZrBf.png',
      activeIcon: 'https://zos.alipayobjects.com/rmsportal/Zral0spCWaHCcCoWxZZk.png',
      redDot: '中文'
    },
    {
      name: 'tab4',
      tag: 'tab44',
      icon: 'https://zos.alipayobjects.com/rmsportal/xBa0vobgYpocSlwanZVM.png',
      activeIcon: 'https://zos.alipayobjects.com/rmsportal/MTikkHhrAneFXlrfARGD.png',
      redDot: ' '
    },
    {
      name: 'tab5',
      tag: 'tab55',
      icon: 'https://zos.alipayobjects.com/rmsportal/AukzoUbVCWPJqSjEbFLu.png',
      activeIcon: 'https://zos.alipayobjects.com/rmsportal/RcumdvGStyfBEDaCuqCt.png',
      redDot: '1234567890'
    }
  ]
});
```

## 2. 代码演示

### 2.1. 基本使用

```
<script>
  var base64 = {
    activeIcon: 'iVBORw0KGgoAAAANSUHEUgAAADQAAAAvCAQAAAAGYCE4AAAABGdBTUEAALGPC/xhBQAAACBjSFJNAAB6JgAAgIQAAAPoAAACA6AAAdTAAA0pgAAA6mAAAF3CcuLE8AAA
AAmJLR0QA/4ePzL8AAAAHdELNRQfeBAoWAhU5ncjwAAACMELEQVRYw72XT0gUURzHP2/bbDV1q1Na9geC8LJSGkSXhMJrQR28Raf0YtBRunct1Dpq0CGyDh7skEVkGRQUgWwhFJGLh421Zbw
d3X0dGsd947DNa+c33x8M8x3m9/vMb+bNezNgq3N8Qv0T2+y3zrVQmnM0G50k5ECnKXigIoPhEx0WoON0ePu70SsHakMZ2ND5tqBWw3WxSwp01HAqfKIdqIUjhs/jyIA002f4HBUZ0Hl6DP8
tfKoNKM2w4SssWV1maF1iw3tZNZoVeiUw+3hhYDSPww9uG41SNTBVRkpgzvDD1897uqLHH0SLD+NwLXpMN098GM0s7VFjDjCzDa0ZtgX9a7bq51bgqv0b1zxjiZxxNLGtXoIqq3ym3AiS4gr
LAd1sRYWSEeuBSCO9ehSwkzQ730LrrLLX0s8oFyN7U6YUp7h0L0n3wBrTj0Nwghk0RQQBKMfb3+34xRAw6Jtumo4EAz5208cATS3CfqAWNhrSBF/tWC7lP+fPnIj2XyVhIrygafcJysNyjL
CIth/BdnqJHfJxAGCDBfiAUf3XKBqPKAa7wCkR12ZBzyUBuW4wxsWKUId2sgzt2kkn1EnY1vf6rKDoY098YDqJAv6yvfnXanBUGSF50zyRRr0iJss1y+hUrfuMsPmSi0FamXI/IMPAln8aze
Q4+9owXdCgSygmu71FRv1NskIY2S8skXGmQdKZEmh6GQPLb4SFco41ACFcrfKc6DIM8eEmfQH27n1x37W3T4AAAAldEVYdGRhdGU6Y3JlYXRlADlwMTYtMDktMTdUMTU6MTk6NTErMDg6MDA
6DCD0AAAAJXRFWHRkYXRlOm1vZGlmeQAYMDE0LTA0LTEwVDIyOjAyOjIxKzA4OjAwnWo3LwAAAE10RVh0c29mdHdhcmUASW1hZ2VNYWdpY2sgNy4wLjEtNiBRMTYgeDg2XzY0IDIwMTYtMDk
tMTcgaHR0cDovL3d3dy5pbWFnZW1hZ2ljay5vcnZlZ2V0AAAAAGHRFWHRUaHVtYjo6RG9jdW1lbnQ60lBhZ2VzADGn/7svAAAAGXRFWHRUaHVtYjo6SW1hZ2U60khlaWdodAAxMDAyPHMfCgA
AABh0RVh0VGh1bWI60klTYWdl0jpXaWR0aAAxMTAyKPIpJAAAABl0RVh0VGh1bWI60k1pbWV0eXBlaGltYWdlLL3BuZz+yVk4AAAAAXdEVYdFRodW1i0jpNVGltZQAxMzk3MTM4NTQxZnpipagA
AABJ0RVh0VGh1bWI60lNpemUAMTkuNEtCujeYBQAAAF90RVh0VGh1bWI60lVSSQBmaWxl0i8vL2hvbWUvd3d3cm9vdC9zaXRLL3d3dy5lYXN5aWNvbi5uZXQvY2RuLWltZy5lYXN5aWNvbi5
jbi9zcmMvMTE0MzgvMTE0MzgyMS5wbmf20jDHAAAAAE1FTkSuQmCC',
    icon: 'iVBORw0KGgoAAAANSUHEUgAAADEAAAAAwCAYAAAC4wJK5AAAABGdBTUEAALGPC/xhBQAAACBjSFJNAAB6JgAAgIQAAAPoAAACA6AAAdTAAA0pgAAA6mAAAF3CcuLE8AAAABmJLR
0QA/wD/AP+gvaeTAAAAB3RJTUUh3gQdFRYXNrWv9gAABL9JREFUaN7V2XmIlGUcB/DP6HpEeVeaFpLL0R1GB51UFiVG2EWRfUEKhUdCxpCdJBEPagdlLBREURERwRpZkJgUVaUZViYWoFwd
muax+50f/yecd8dd93ZddfZ+cLLvjPMPu/v+/705/vkVAiFfG3x9LLch2PwKx7FbNRBbsWMVteqqTCBU5PBB2ITBuN+/Ic55a7XrRIkEmowKRGYh3F4Anvj3kQwS7hLkhiM07AZ07EEU/EG9
sdk9C5noUqSGJaM/QWr0ncb8Qj+xBgC3SVJZMLjY0yDtcnoIpZhKfrjpC5JIIgHsen+CxFSRWzR6JlRXY5Exgt5nC+q0LwWSEJDlyKRIZDD9SInluAajmvSD3jg03a/oUiQyGIsbhBeeFP0hS
3Ikjhd5srTLkMgY0ALTRNK+ivnN/Pwi7IfP8F2XIJEhMASzcIJI5gexreTnQzEBBbysacLveRKffG0pgZkYL0rqZKxmp9loAo7CV1hQ7rP2RDinwrPJwHW4GYuyZBNGYgK6f0YMG5UbAD0GD
cfz0Bk/4ja81cy/9EatDhcV65W2PK8zPdEddycCy3FFlkDJG74E14mx4yH80cxvWkSHeyLjhtG4Mhk2FZ+0QKA/bsdeYi/xbjNrNUEpuc7aT/TAjegr8uGdlgwQM1Qe6/GSiI4ezaxZj+2l5
HIrZnQsiczip4ux4i88lwXoKTz+1bghekrMTqXICY+ux0f4QBQJhXxtx5HIE0gu4rsfXpDCaBfx/YPoCXeJ/UU5WIIYZeA1ba8rZ0bWGEgPH4WJsEJWpvpV/r8cDY0YaJhpds4/BkTgbx4oyf
BAezhXytTUYLep5uZ7ZkB66vuT74Xhdd0XHcSfqd1Vl2vESh+JW3CHEhJtq0ofJYl4pFw1YjGtFByaqzPRE4G0xQ2vNC2WX0QzZdbHhH0skTMwV8rW/YyC+xD8aZ/kW1xNuHyk29vNFFRqPq
0RTuxoftsXIcpEhc67o0w25Qr62kIy/AJ+KxNwVtqc3MDt9btDYNNeJ0WHHRqeJWSiHIIf3MaCYAwVR2raVuc72jNErhfe2JWJ7ggAcgj74LZvIuXIeXJKIb4t5qJvwyNY9RGCQyMcavLe7f
aIua3hnEsign9gRjhfvCw6HNLt0NroUm0QDvVBEQK6S4ll7USeq4gLSiz0rikTG49vxc7rPVxWJEhRt31gRab89KKmKA8WoBN9UDYmEHAZgiph412BxtZE4QpTXs8RcNgfLqy0nVotZrQ5/Y
yGVPZ9oD7ZhrhdfBon9RfwQKcmvRWwWb1WRyFSn/jhAhNRPVU0ipLyegsPE3LSKCh0BtxM9xYnqFCHpLBIltjo8kdALtwhPrBXzUx2774kGdhaz0gkb8SYux/f4uvi83fXEaKF2Dyl+0RES0
C7QM/1tcpaX3Z7WtcGI4iJnpGuhEMzWtWGNtqK70JzgWxmLsEiip1Cv+2g9T+pF+ydcvAbnCLVvrkYtqpAhW3oK2qCpSFaws8rSLV3d03UeLTPMiWuN0Bs7EY8pQydKKAq+M8UsM1uE1RjRj
IoGFjJXFfi19zhLJpatIpJeIlLLC5diRfzVC0JuWPNFTedgilL6n05uvFargNUKi72hsFnLSi0IabbKv/x9eolhQ3qHKMQAAACV0RVh0ZGF0ZTpjcmVhdGUAMjAxNi0wOS0xN1QxNToyMDoy
Csw0DowMMcmIfUAAAAldEVYdGRhdGU6bW9kaWZ5ADIwMTYtMDktMTdUMTU6MTk6NTErMDg6MDBhsqZTAAATXRFWHRzb2Z0d2FyZQBjbWFnZU1hZ2ljayA3LjAuMS02IFExNiB4ODZfNjQgM
jAxNi0wOS0xNyBodHRwOi8vd3d3Lm1tYWdlbWFnZW50Lm9yZ93ZpU4AAAAAYdEVYdFRodW1i0jpEb2N1bWVudDo6UGFnZXMMAMaf/uy8AAAAAYdEVYdFRodW1i0jpJbWFnZTo6SGVpZ2h0ADc2M
C1GcAQAAAAAXdEVYdFRodW1i0jpJbWFnZTo6V2lkdGgANzgyZjps+wAAABl0RVh0VGh1bWI60k1pbWV0eXBlaGltYWdlLL3BuZz+yVk4AAAAAXdEVYdFRodW1i0jpNVGltZQAxMzk4Nzc3NzQzr
uA5SgAAABJ0RVh0VGh1bWI60lNpemUAMjAuNEtCMb0I2gAAAF90RVh0VGh1bWI60lVSSQBmaWxl0i8vL2hvbWUvd3d3cm9vdC9zaXRLL3d3dy5lYXN5aWNvbi5uZXQvY2RuLWltZy5lYXN5a
WNvbi5jbi9zcmMvMTE1NjkvMTE1Njk4NS5wbme768jyAAAAAE1FTkSuQmCC'
  };

function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}

ready(function(){
  AlipayJSBridge.call('tabBar', {
    action: 'create', // 创建 Tab Bar, 必选
    textColor: 16776960, //ffff00 黄, 必选
  });
});
```

```

selectedColor:16711680, //ff0000 红, 必选
selectedIndex:1, // 必选
items: [ // 所有 TabItems, 必选
  {
    name: 'tab1', // 标题, 必选
    tag: 'tab11', // tab项id, 必选
    icon: 'https://zos.alipayobjects.com/rmsportal/I0wBUWsF0QyNYhhAxDgb.png', // 正常状态图片 URL, 支持 base64 编码 URL, 必选
    activeIcon: 'https://zos.alipayobjects.com/rmsportal/KdyZNLUKgFfAAHreQzpj.png', //选中状态图片 URL, 支持 base64 编码 URL, 必选
    redDot: -1, // 选填, '0' 表示红点, null、'-1' 表示不显示, 其他值展示出来
  },
  {
    name: 'tab2',
    tag: 'tab22',
    icon: base64.icon, //'https://zos.alipayobjects.com/rmsportal/dMHfRGFxWiRidgHSLLvS.png', // 灰色
    activeIcon: base64.activeIcon, //https://zos.alipayobjects.com/rmsportal/tNPTHyWzgEQNlZLxHVRL.png', // 选中状态图片 URL, 支持 base64 编码
URL, 必选
    redDot: '99+'
  },
  {
    name: 'tab3',
    tag: 'tab33',
    icon: 'https://zos.alipayobjects.com/rmsportal/neFZyipaNvaZiMgNZrBf.png',
    activeIcon: 'https://zos.alipayobjects.com/rmsportal/Zral0spCWaHCcCoWxZZk.png',
    redDot: '中文'
  },
  {
    name: 'tab4',
    tag: 'tab44',
    icon: 'https://zos.alipayobjects.com/rmsportal/xBa0vobgYpocSlwanZVM.png',
    activeIcon: 'https://zos.alipayobjects.com/rmsportal/MTikkHhrAneFXlrfARGD.png',
    redDot: ' '
  },
  {
    name: 'tab5',
    tag: 'tab55',
    icon: 'https://zos.alipayobjects.com/rmsportal/AukzoUbVCWPJqSjEbFLu.png',
    activeIcon: 'https://zos.alipayobjects.com/rmsportal/RcumdvGStyfBEDaCuqCt.png',
    redDot: '1234567890'
  }
]
});

```

// 修改某个下标的badge

```

document.addEventListener('tabClick', function(e){
  console.log('tab clicked! - ' + e.data.tag); // data就是tag的值, 请前端自己处理
  console.log(e);
  AlipayJSBridge.call('tabBar', {
    action: 'redDot',
    tag: e.data.tag,
    redDot: '12' // -1、0、或者 null 表示取消
  });

```

```
});
});
});
</script>
```

### 3. API

```
AlipayJSBridge.call('tabBar', {
  action: , // 创建 Tab Bar, 必选
  textColor: , //ffff00 黄, 必选
  selectedColor: , //ff0000 红, 必选
  selectedIndex: , // 必选
  items: [ // 所有 TabItems, 必选
    {
      name: , // 标题, 必选
      tag: , // tab项id, 必选
      icon: , // 正常状态图片 URL, 支持 base64 编码 URL, 必选
      activeIcon: , //选中状态图片 URL, 支持 base64 编码 URL, 必选
      redDot: // 选填, '0' 表示红点, null、'-1' 表示不显示, 其他值展示出来
    }
  ], fn)
```

#### 3.1. 入参

| 名称            | 类型       | 描述   | 必选 | 默认值 | 版本     |  |
|---------------|----------|--|----|-----|--------|--|
| action        | string   | 操作类型，目前支持’create’和’redDot’，’create’创建tabBar，’redDot’修改红点展示 | Y  |     | 10.0.0 |  |
| textColor     | number   | 默认tab项标题颜色   | N  |     | 10.0.0 |  |
| selectedColor | number   | tab项标题选中颜色   | N  |     | 10.0.0 |  |
| selectedIndex | number   | 默认选中tab项索引   | N  |     | 10.0.0 |  |
| items         | array    | tab项   | Y  |     | 10.0.0 |  |
| name          | string   | tab项标题   | Y  |     | 10.0.0 |  |
| tag           | string   | tab项id   | Y  |     | 10.0.0 |  |
| icon          | string   | tab项图标，支持base64  | Y  |     | 10.0.0 |  |
| activeIcon    | string   | tab项选中时图标，支持base64，但不要加前面的数据格式头「data:image/png;base64,」    | Y  |     | 10.0.0 |  |
| redDot        | string   | ‘0’表示红点，’-1’表示不显示，其他值展示出来                                  | N  |     | 10.0.0 |  |
| fn            | function | 回调函数   | N  |     |        |  |



Since 8.0

# toast

显示一个弱提示，可选择多少秒之后消失 安卓需要注意，如果用户关闭系统通知，则此弹框不会出现

## 1. 使用方法

```
AlipayJSBridge.call('toast', {
  content: '操作成功',
  type: 'success',
  duration: 2000
}, function(){
  alert("toast消失后执行");
});

// 可以通过hideToast接口隐藏已经弹出的toast

AlipayJSBridge.call('hideToast', {
}, function(){
});
```

## 2. 代码演示

### 2.1. 各种类型演示

```
<h1>点击以下按钮看不同效果</h1>
<a href="javascript:void(0)" class="success">显示成功信息</a>
<a href="javascript:void(0)" class="fail">显示失败信息</a>
<a href="javascript:void(0)" class="exception">显示异常信息</a>
<a href="javascript:void(0)" class="none">只显示信息</a>
<a href="javascript:void(0)" class="duration">3.5s信息显示</a>

<script>
function toast(config, callback){
  AlipayJSBridge.call('toast',config, callback);
}

function ready(callback) {
```

```
// 如果jsbridge已经注入则直接调用
if (window.AlipayJSBridge) {
    callback && callback();
} else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
}
}
ready(function(){
    document.querySelector('.success').addEventListener('click', function(){
        toast({
            content: '操作成功',
            type: 'success'
        });
    });

    document.querySelector('.fail').addEventListener('click', function(){
        toast({
            content: '网络繁忙, 请稍后再试',
            type: 'fail'
        });
    });

    document.querySelector('.exception').addEventListener('click', function(){
        toast({
            content: '发生异常, 请注意',
            type: 'exception'
        });
    });

    document.querySelector('.none').addEventListener('click', function(){
        toast({
            content: '欢迎光临',
        });
    });

    document.querySelector('.duration').addEventListener('click', function(){
        toast({
            content: '请稍等',
            duration: 3500,
        }, function(e){
            alert('toast消失回调');
        });
    });
});
});
</script>
```

### 3. API

```
AlipayJSBridge.call('toast',{
  content, type, duration
}, fn)
```

### 3.1. 入参

| 名称       | 类型       | 描述  | 必选 | 默认值    | 版本  |
|----------|----------|---|----|--------|-----|
| content  | string   | 文字内容，如果使用antbridge版本，请用text属性替代                                     | Y  | ‘’     | 8.0 |
| type     | string   | none / success / fail / exception。 excption类型必须传文案                  | N  | ‘none’ | 8.1 |
| duration | int      | 显示时长，单位为毫秒。android实际只支持2000/3500两种，小于或等于2000的相当于2000，大于2000的相当于3500 | N  | 2000   | 8.1 |
| fn       | function | 回调函数，当toast消失后被调用   | N  |        |     |

## 4. 使用注意

- toast 虽然会自动关闭，但是也可以通过 hideLoading 来关闭，这种使用方式不常见，但是要防止被hideLoading关闭。
- 安卓需要注意，如果系统通知关闭，则此弹框不会出现。
- 特别注意antbridge版本的toast，要求传入的是 text 属性，而不是 content 属性。

Since 9.9

## getServerTime(获取服务器时间)

通过APTTimeUtilGetServerTimeWithStrictMode方法来获取当前服务器时间的毫秒数，如果离线情况那么为-1

### 1. 使用方法

```
AlipayJSBridge.call('getServerTime', function (data) {
  console.log(data.time)
});
```

### 2. 代码演示

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  AlipayJSBridge.call('getServerTime', function (data) {
    alert(data.time)
  });
});
</script>
```

### 3. API

getServerTime

#### 3.1. 入参

| 名称 | 类型       | 描述   | 必选 | 默认值 | 版本 |
|----|----------|------|----|-----|----|
| fn | function | 回调函数 | N  |     |    |

#### 3.2. 出参

回调函数带入的参数result: {time: string}

| 名称          | 类型     | 描述  | 备注 |
|-------------|--------|-----|----|
| result.time | string | 毫秒数 |    |

Since 8.1

# setSessionData

## 1. 使用方法

```
AlipayJSBridge.call('getSessionData', {
  keys: ['name']
}, function (result) {
  console.log(result.data.name);
});
```

## 2. 代码演示

### 2.1. session数据读写

<h2>点击下面按钮来查看效果</h2>  
<a href="javascript:void(0)" class="set">设置session数据</a>  
<a href="javascript:void(0)" class="get">读取session数据</a>

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.set').addEventListener('click', function(){
    AlipayJSBridge.call('setSessionData', {
      data: {
        name: 'Liu Hunan',
        age: '15'
      }
    });
  });

  document.querySelector('.get').addEventListener('click', function(){
    AlipayJSBridge.call('getSessionData', {
      keys: ['name']
    }, function (result) {
      alert(JSON.stringify(result));
    });
  });
});
</script>
```

2.2. 入参

| 名称   | 类型       | 描述         | 必选 | 默认值 | 版本 |
|------|----------|------------|----|-----|----|
| keys | array    | 要读取的数据的key | Y  |     |    |
| fn   | function | 回调函数       | N  |     |    |

2.3. 出参

回调函数带入的参数result: {data}

| 名称   | 类型     | 描述          |
|------|--------|-------------|
| data | object | keys对应的字段数据 |

### 3. 使用注意

- sessionData 是以每个app为单位的，因此可以实现app内的数据共享。
- 当app退出之后，数据自动清除。
- 没有办法去remove sesionData 的key，只有通过设置一个空值来清除。
- 假如用户账号在其他设备登入导致logout，那么并不会清除数据，这个时候如果这个设备登入另一个账号，是可以看到这个数据的。

Since 8.1

## setSessionData

会话数据写入

### 1. 使用方法

```
AlipayJSBridge.call('setSessionData', {
  data: {
    name: 'Liu Hunan',
    age: '15'
  }
});
```

### 2. 代码演示

#### 2.1. session数据读写

<h2>点击下面按钮来查看效果</h2>  
<a href="javascript:void(0)" class="set">设置session数据</a>  
<a href="javascript:void(0)" class="get">读取session数据</a>

```
<script>
function ready(callback) {
  // 如果jsbridge已经注入则直接调用
  if (window.AlipayJSBridge) {
    callback && callback();
  } else {
    // 如果没有注入则监听注入的事件
    document.addEventListener('AlipayJSBridgeReady', callback, false);
  }
}
ready(function(){
  document.querySelector('.set').addEventListener('click', function(){
    AlipayJSBridge.call('setSessionData', {
      data: {
        name: 'Liu Hunan',
        age: '15'
      }
    });
  });

  document.querySelector('.get').addEventListener('click', function(){
    AlipayJSBridge.call('getSessionData', {
      keys: ['name']
    }, function (result) {
      alert(JSON.stringify(result));
    });
  });
});
</script>
```

2.2. 入参

| 名称   | 类型         | 描述  | 必选 | 默认值 | 版本 |
|------|------------|---|----|-----|----|
| data | dictionary | 要写入的数据，每个key/value pair为一个数据项。value只支持字符串 | Y  |     |    |

3. 使用注意

- sessionData 是以每个app为单位的，因此可以实现app内的数据共享。



- 当app退出之后，数据自动清除。
- 没有办法去remove sessionData 的key，只有通过设置一个空值来清除。
- 如果你需要在app之间共享数据，或者需要做持久化，那么可以使用 sharedData 。
- 假如用户账号在其他设备登入导致logout，那么并不会清除数据，这个时候如果这个设备登入另一个账号，是可以看到这个数据的。
- 特别注意的是，在安卓下，你存进去的内容，取出来之后类型并不能保证不变。比如，存进去一个number: 200，getSessionData获取的值会变成”200”。