

# 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

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； 无效的index； 未匹配url； 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 时所带的数据如何被接收？ 请查看resume事件 (../events/resume.html)

