

# 触宝开放平台JS-SDK文档

- [概述](#)
- [接口列表](#)
- [基础接口](#)
- [登陆接口](#)
- [支付接口](#)
- [定位接口](#)
- [全屏接口](#)
- [回退刷新接口](#)
- [其他信息](#)

## 概述

- 触宝JS-SDK是触宝开放平台面向O2O服务商提供的基于触宝电话内生活服务网页开发工具包
- 通过JS-SDK，O2O服务商可以借助触宝电话获取到在自身网页内唤起登陆，发起支付，进行定位等功能
- 通过SDK，可以具有更强大的能力，做到在触宝电话内更加一致的体验，打通触宝用户和O2O服务

## 接口列表

- ready
- checkJsApi
- logged
- login
- show
- dismiss
- select
- pay
- getLocation
- fullScreenLoadPage
- backWithRefresh

## 基础接口

页面加载完需要立刻调用接口（仅针对iOS）

```
ctk.ready(function(){
    //如果需要在页面加载完就调用相关接口，则须把相关接口放在ready函数中调用来确保正确执行。
    //对于用户触发时才调用的接口，则可以直接调用，不需要放在ready函数中。
});
```

## 判断网页打开环境是否支持接口

```
ctk.checkJsApi({
    jsApiList: ['logged', 'login'],//需要判断是否支持的api名称列表
    success: function(res) {
        //判断支持之后之后的回调函数，res是结果字典，包含errMsg和result
        //示例: {"errMsg": "check_success!", "result" : {"logged" :
true, "login" : true}}
    }
});
```

# 登陆接口

## 判断用户是否已经登陆

```
ctk.logged({
    yes: function(res) {
        //判断用户已登录的回调函数，用户的accessToken存放在res.accessToken
        内
    },
    no: function(res) {
        //判断用户为未登陆的回调函数，建议在这个回调函数中唤起登陆框，见下一个
        接口
    }
});
```

## 唤起应用内登录框

```
ctk.login({
  phone: '18512345678', //填入登录框的手机号
  success: function(res) {
    //唤起登录框用户登陆成功后的回调函数,用户accessToken存放在res.accessToken内;
    //如果需要得到用户信息, 建议在这个回调函数内写一个request(accessToken)方法获取用户手机号和UserId
    //拿用户手机号和UserId需要在服务商后台实现接口详见touchlife.api.pdf
  },
  fail: function(res) {
    //唤起登录框用户登陆未成功的回调函数
  }
});
```

## 支付接口

### 展示支付过渡框

```
ctk.show(); //建议用户点击支付跳转时候使用, 使得体验更加一致
```

### 关闭支付过渡框

```
ctk.dismiss();
//关闭支付过渡提示框,一般放在ctk.pay()的finish中执行,
//如果放在ctk.select()的success中执行, 支付宝弹出前提示框会消失, 影响用户体验
//具体参见demo页代码
```

### 唤起支付选择界面

```

ctk.select({
    tradeService: 'com.company.product', //服务商在触宝开放平台的标
    识, 格式为com.公司.产品, 用于获取优惠信息
    totalFee: 10000, //支付的金额, 单位为分
    <products>: [STRING, ...], //商品列表, 用于特定商品优惠, 可选参数

    success: function(res) {
        //用户点击支付页面支付按钮成功之后的回调函数
        var tradeStr = res.tradeStr; //支付信息的字符串
        requestPay(tradeStr); //后台获取加密支付信息的请求
        self.show(); //获取加密字符串的同时显示加载提示, 减少用户焦虑
    },
    fail: function(res) {
        //用户退出支付选择页面的回调函数
    }
});

```

备注: requestPay由服务商自己实现, 主要功能是拿着tradeStr向自己服务器请求加密的支付信息字符串, 可见

<http://test.open.cootekservice.com:15088/> 网页demo源码

### 调用本地支付的接口

```

ctk.pay({
    payStr: [payStr], //获取到的加密支付信息
    finish: function(resCode) {
        //支付完成之后的回调函数, resCode是支付状态码, 包含成功和失败等状态
    }
});

```

备注: 此接口一般在请求加密字符串成功的回调函数中进行, 即上个接口中的requestPay中, 加密支付信息需要到服务商服务器去获取, 具体相关接口为/trade/create, 支付流程较复杂, 可查看api文档了解支付流程, 也可查看

<http://test.open.cootekservice.com:15088/> 网页demo源码

## 定位接口

### 调用本地定位接口



```
ctk.getLocation({
  success: function(res) {
    //定位之后的回调函数，res包含四个key:city,address,latitude,longitude,
    //分别代表定位城市，文字地址，纬度，经度
    //如{"city" : "上海", "address" : "上海市徐汇区虹梅路2007号", "latitude" : 31.17 , "longitude" : 121.38}
  }
});
```

## 全屏接口

### 全屏接口

```
ctk.fullScreenLoadPage(url, selfAdjust, quitConfirm);
//目的Android、iOS网页跳转时打开一个全屏页面；
//调用举例(访问参数中的url，则会全屏显示页面)
a.仅做全屏页面跳转，不支持横竖屏及退出确认弹框；
    ctk.fullScreenLoadPage(url);
b.支持全屏页面及横竖屏显示；
    ctk.fullScreenLoadPage(url,true);
c.支持全屏页面、横竖屏显示及退出确认弹框。
    ctk.fullScreenLoadPage(url,true,true);
```

## Android回退刷新接口

### Android回退刷新接口

```
ctk.backWithRefresh();
//原因：在Android平台，由于每个页面是一个单独保存的fragment，所以回退时前面的页面不会刷新；
//方案：第三方从A页面跳到B页面，返回时希望A页面重新加载，需要在B页面加载完成后调用backWithRefresh接口，并且在A页面实现一个名称为refreshPage()的函数，该函数里放一些A页面加载的处理逻辑，比如执行一些初始化函数甚至直接调用location.reload()。那么在回退时，客户端会调用A页面的refreshPage函数，达到重刷的目的。
```

## 其他信息

- js-sdk在线引用地址

