

# 触控广告平台 SDK v4.1.1 开发说明文档

## Android 版

文档版本 4.0

## 目录

|                                    |    |
|------------------------------------|----|
| 1. 版本更新说明 .....                    | 3  |
| 2. 概要 .....                        | 3  |
| 3. 广告类型 .....                      | 3  |
| 4. 开发环境: .....                     | 5  |
| 4. 开发步骤: .....                     | 5  |
| 4.1、注册并获取触控广告平台 publisherID .....  | 5  |
| 4.2、在项目中集成触控广告平台 SDK .....         | 5  |
| 4.3、AndroidManifest.xml 文件配置 ..... | 6  |
| 4.4、嵌入广告 .....                     | 7  |
| (1)、嵌入固定广告 .....                   | 7  |
| (2)、嵌入精品推荐广告: .....                | 11 |
| (3)、精品推荐按钮: .....                  | 13 |
| (4)、嵌入插屏广告: .....                  | 14 |
| (5)、积分墙广告: .....                   | 16 |
| (6)、积分墙按钮: .....                   | 20 |
| 5. 常见错误码说明: .....                  | 21 |
| 6. 注意事项: .....                     | 22 |

## 1. 版本更新说明

### 4.1.0->4.1.1

插屏广告支持两种关闭模式，一种是原有的关闭按钮，需要手动关闭（或者点击广告进行关闭），一种是倒计时方式，开发者可以设置 `closeMode(int mode)`，支持设置倒计时时长。

### 4.0.2->4.1.0

增加新的广告类型：积分墙广告，一种完成任务可以返回积分的广告类型。（如果有需要混淆代码的，新增加广告类型相对之前的版本需要新增 `keep` 类，详见本文档的【注意事项】）

### 4.0.1-->4.0.2

内部做了一些优化。

### 4.0.0-->4.0.1

将资源文件打入到 `jar` 包中，不再单独放在 `res` 文件夹下。

### 3.2.3->4.0.0

- a.对代码进行了重构；
- b.优化了下载模块；
- c.修改了广告逻辑，对外接口也相应发生了改变，详见后续开发步骤。

## 2. 概要

触控广告 SDK Android 版 v4. 1. \*(以下简称 SDK) 包解压后将会得到一个 `PunchBoxSDKDemo` 文件夹、一个 `libs` 文件夹、一个 SDK 开发文档：

`PunchBoxSDKDemo` 文件夹包含调用示例工程；

`Libs` 文件夹包含开发所需要的 `jar` 包；

开发文档包含开发流程说明。

## 3. 广告类型

触控广告平台 SDK 提供四种类型的广告：固定广告、精品推荐广告、弹出广告和积分墙广告。

(1) 固定广告：又称 banner 广告，是一种常见的以横条方式展示的广告。效果如下图：



(2)精品推荐广告：以列表的方式推荐热门应用，点击可直接下载。

展现效果如下图：



(3)弹出广告：展示图片、视频或其他多媒体的广告，

展现效果如下图：



(4) 积分墙广告：按照指定的赚取说明完成任务后，给予一定的奖励。

展现效果如下图：



#### 4.开发环境：

触控广告 SDK 支持 Android SDK 2.3 及其以上版本，对低于此版本的手机，PunchBox SDK 支持嵌入，但不展现广告。

#### 4.开发步骤：

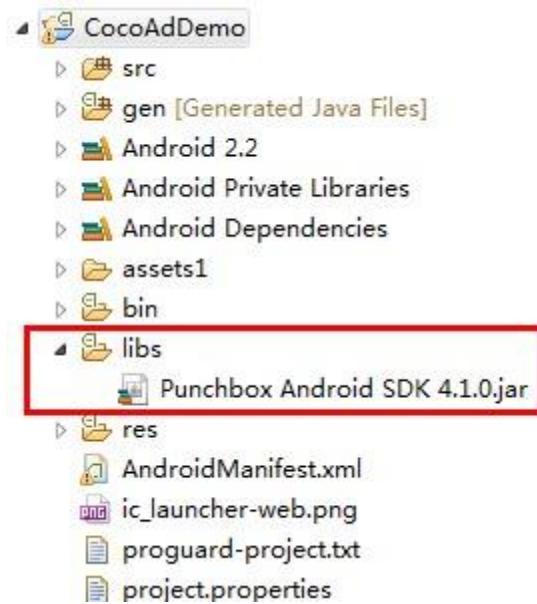
##### 4.1、注册并获取触控广告平台 publisherID

要使用触控广告平台提供的功能，您需要先注册PunchBox帐号，并通过创建应用获取Publisher ID.

注册Publisher ID地址：<http://dev.cocounion.com/>，如您已经有Publisher ID,可以直接进入第二步。

##### 4.2、在项目中集成触控广告平台 SDK

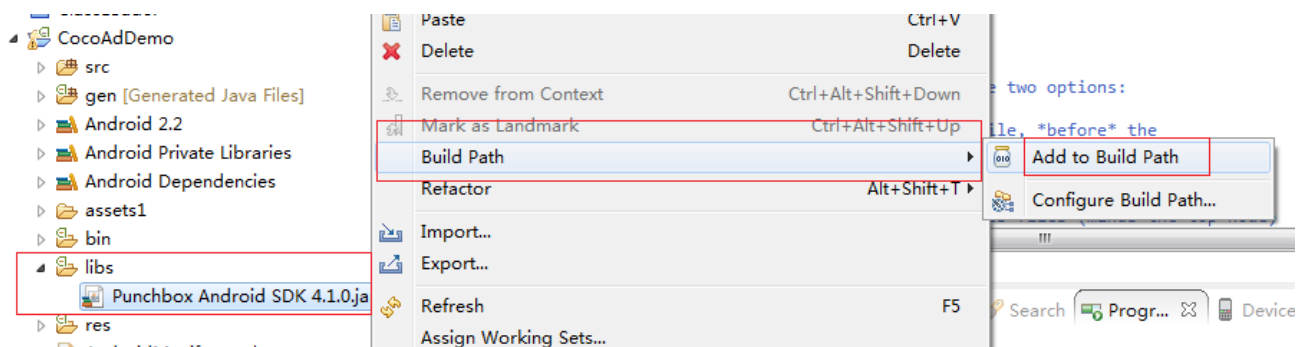
- 1、将获取到的 SDK 开发包解压，libs 文件夹里包含广告 SDK；
- 2、将 libs 目录下的 jar 包引入到应用工程的依赖库中，拷贝 jar 包到 Eclipse 工程的 libs 目录下，如下图：



3. 添加后，刷新工程，点击上图工程的“Android Private Libraries”目录，打开，如下面自动出现刚引入的 jar 文件，说明引入成功，如下图：



如未出现，可以手动再添加：右键点击 libs 下刚拷贝的 jar 文件，选择弹出菜单里的“Build Path”，然后选择“Add to Build Path”，如下图：



#### 4.3、AndroidManifest.xml 文件配置

##### 1、增加权限

(1). PunchBox SDK 需要添加如下 permission:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

## 2、增加 Service 声明

请在配置中间增加如下内容:

```
<service android:name="com.punchbox.engine.PunchBoxService"
    android:process="com.punchbox.punchboxservice"
    android:exported="true">
    <intent-filter>
        <action android:name="com.punchbox.service"/>
    </intent-filter>
</service>
<meta-data android:name="chukong_publisherid"
    android:value="100011-0A2E90-7CDB-6FD3-A9B983ABBBBA"/>
```

注意:

从 3.2.3 开始需要在 AndroidManifest.xml 里配置 chukong\_publisherid, 修改 chukong\_publisherid 对应的值, 如下:

```
<meta-data android:name="chukong_publisherid" android:value="修改成你申请的publisherid"/>
```

## 4.4、嵌入广告

### (1)、嵌入固定广告

a. 通过 xml 布局文件嵌入固定广告位

- 在 Demo 工程中 res/layout/activity\_banner.xml (此处假设是您将要展示固定广告的 activity 所配置的 xml 文件) 中定义 com.punchbox.ads.AdView, 如果指定其使用 ads:loadAdOnCreate 属性表示应用启动立即加载广告。

- 除了强制 `AdView` 立即加载广告之外，您还可以选择在运行时中将 `AdView` 作为资源来查询，并要求其请求广告。

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:ads="http://schemas.android.com/apk/lib/com.punchbox.ads"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <com.punchbox.ads.AdView android:id="@+id/adView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        ads:placementID="您的广告位id"
        ads:loadAdOnCreate="true"/>
</LinearLayout>
```

如果希望在 xml 中支持 `ads:loadAdOnCreate` 和 `ads:placementID` 属性，需要在工程的资源文件夹 `res/xml` 下创建一属性文件,假设名称为 `attrs.xml`,内容如下(也可从 Demo 中该目录下拷贝):

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <declare-styleable name="com.punchbox.ads">
        <attr name="placementID" format="string"/>
        <attr name="LoadAdOnCreate" format="boolean" />
    </declare-styleable>
</resources>
```

注意，您必须将广告位 ID 替换为您的广告位 ID（并非 Publisher ID），如果没有，为空也行。在 Android 布局创建 `AdView` 后，这段代码就会立即尝试加载广告。

#### 提示:

什么是广告位 ID？它的作用是什么？

广告位是用来区分在同一个应用中，不同位置接入同一种广告类型的标识。如果开发者不同位置广告传入了不同广告位 id，最后能在平台上看到不同展示位置对应分开和合计的广告数据报表，帮助开发者更多了解用户的行为。如果开发者不希望区分，传空或者 XML 中不设置即可。

广告位 ID 是有别于 PunchBox ID 的。

广告位 ID 需要开发先在平台进行创建，然后才能在 SDK 中传入，如果没有先创建，传入将无法识别。



如果您不希望自动加载广告，需要控制用于在应用中加载广告的 `AdRequest`，可以从上述代码中删除 `ads:loadAdOnCreate="true"` 这一行，改为以查询资源的方式通过 `findViewById` 查询 `AdView`，并调用 `loadAd` 来加载广告：

```
import com.punchbox.ads.*;

public class BannerXMLActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_banner);

        // AdView 加载广告。
        AdView adView = (AdView) this.findViewById(R.id.adView);
        adView.loadAd(new AdRequest());
    }
}
```

b.通过代码方式创建固定广告：

- 声明 `AdView` 实例
- 指定广告位 ID，完成创建过程
- 将该视图添加进用户界面
- 设置监听(可选)
- 对其加载广告

为了方便起见，最好在应用的 `Activity` 中执行上述所有步骤。

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.LinearLayout;

import com.punchbox.ads.*;
import com.punchbox.exception.PBException;
import com.punchbox.listener.AdListener;

public class BannerCodeActivity extends Activity implements AdListener{
    private AdView adView;
    private String placementID = "abcd";//您的广告位ID

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_banner_code);

        // 创建 adView，如果不传入placementID，可以用另一个构造函数AdView(context)
        adView = new AdView(this, placementID);
    }
}
```

```
// 查找 LinearLayout, 假设其已获得
// 属性 android:id="@+id/mainLayout"
LinearLayout layout = (LinearLayout)findViewById(R.id.mainLayout);

// 在其中添加 adView
layout.addView(adView);

// 启动一般性请求并在其中加载广告
adView.loadAd(new AdRequest());
}

@Override
public void onDestroy() {
    adView.destroy();
    super.onDestroy();
}

//以下是监听广告回调部分
@Override
public void onReceiveAd() {
    //对于固定广告在收到广告后回调, 对于有预加载的弹出和精品推荐广告, 仅仅是在预加载成功
    //后回调, 但不一定是会展示
}

@Override
public void onFailedToReceiveAd(PBException ex) {
    //在请求广告失败后回调
}

@Override
public void onPresentScreen() {
    //展示广告时回调
}

@Override
public void onDismissScreen() {
    //广告移除时回调
}
}
```

## (2)、嵌入精品推荐广告：

步骤：

- 声明 `MoreGameAd` 实例
- 设置监听(可选)
- 对其加载广告
- 在需要展示的时机展示精品推荐；

```
import com.punchbox.ads.AdRequest;
import com.punchbox.ads.MoreGameAd;
import com.punchbox.exception.PBException;
import com.punchbox.listener.AdListener;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;

public class MoreGameActivity extends Activity implements AdListener{

    MoreGameAd ad;
    String placementID = "你的广告位ID";//没有可以传空

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_moregame);

        //声明MoreGameAd实例
        ad = new MoreGameAd(this);
        //设置监听，本类继承了AdListener
        ad.setAdListener(this);
        //加载广告
        ad.loadAd(new AdRequest());
    }
    //通过xml布局文件中配置的按钮触发
    public void moregame(View v){
        try {
            ad.showFloatView(this, 0.9, placementID);
        } catch (PBException e) {
            //当设置的scale不在范围内，或者广告还未加载完毕
            e.printStackTrace();
        }
    }
}
```

```

//以下监听回调
@Override
public void onReceiveAd() {
    // 对于固定广告在收到广告后回调, 对于有预加载的弹出和精品推荐广告, 仅仅是在预加载成功后回调, 但不一定会展示
}

@Override
public void onFailedToReceiveAd(PBException ex) {
    // 在请求广告失败后回调
}

@Override
public void onPresentScreen() {
    // 展示广告时回调
}

@Override
public void onDismissScreen() {
    //广告移除时回调
}
}

```

#### 提示:

为了更快地展示精品推荐, 减少因网络原因带来的延迟, 建议将广告实例化和加载部分放在整个应用启动部分, 整个应用运行过程当中, 只需要调用一次 `loadAd()`, 并不需要每个页面都调用 `loadAd()`。当广告加载完成, 会在回调函数 `onReceivedAd()` 中收到通知, 开发者也可以在此函数中启动 `showFloatView()`。

#### 接口说明:

| 接口名   | 功能描述                       | 参数说明  |
|---|----------------------------|---|
| <code>showFloatView(Activity ac, double scale, String placementID)</code> | 以浮动窗的方式在当前 Activity 展示精品推荐 | 1.activity :要展示精品推荐的 Activity 对象;<br>2.scale: 精品推荐显示窗口相对屏幕大小的比例, 必须大于 0.5, 小于等于 1.0<br>3. <code>placementID</code> : 广告位 ID, 参见【嵌入精品推荐广告】中对此的定义。没有可以传 |

|  |                         |  |
|--|-------------------------|--|
|  |                         | 空。   |
| <code>show(ViewGroup parent,String placementID)</code> | 给定一个容器，精品推荐将展示在此容器内     | 1.parent: 装载精品推荐的容器，可以是各种布局。<br><br>2. <code>placementID</code> :广告位 ID，参见【嵌入精品推荐广告】中对此的定义。没有可以传空。 |
| <code>dismiss()</code>                                 | 从当前屏幕上移除广告，跟点击关闭按钮效果一致。 |  |
| <code>destroy()</code>                                 | 销毁                      |  |

### (3)、精品推荐按钮：

Punchbox SDK 还提供一种一键封装精品推荐的按钮，开发者只需在应用中嵌入该按钮，无需编写代码即可实现打开精品推荐。

在 xml 布局中，嵌入 MoreGameButton，如下：

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity" >

    <com.punchbox.ads.MoreGameButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="您需要替换的图片资源，如果不需要替换，将采用默认的图片，请删除本行"
        ads:placementID="请替换成您的广告位id，如果没有，去掉此行"/>

</LinearLayout>
```

无需编写任何代码，直接封装了打开精品推荐功能,同时支持在 xml 中设置按钮图片(xml 设置图片可以使用:android:src=""属性指定图片,如果采用默认图片，删除该行)。

开发者也可以在代码中通过实例化 MoreGameButton 对象来创建按钮，支持设置按钮图片和精品推荐展示 scale 值。

接口说明：

| 接口名                                    | 功能描述   | 参数说明                     |
|--|--------|--------------------------|
| <code>setImageBitmap(Bitmap bm)</code> | 设置按钮图片 | bm :Bitmap 类型，需要设置为按钮的图片 |

|                                      |  |                                   |
|--------------------------------------|--|-----------------------------------|
| setImageDrawable(Drawable drawable)  | 设置按钮图片                                     | drawable: Drawable 类型，需要设置为按钮的图片  |
| setImageResource(int resId)          | 设置按钮图片资源                                   | resID:int 类型，需要设置为按钮图片的资源 id      |
| setAdListener(AdListener adListener) | 设置广告监听                                     | adListener:AdListener 类型          |
| setScale(float scale)                | 设置点击按钮后，弹出的精品推荐视图相对于屏幕大小，如果 scale 为 1，将填满。 | scale: float 类型，必须大于 0.5，小于等于 1.0 |

#### (4)、嵌入插屏广告：

步骤：

- 声明 InterstitialAd 实例
- 设置监听(此步骤可选)
- 对其加载广告
- 在需要展示的时机展示插屏广告；

```

• import com.punchbox.ads.AdRequest;
• import com.punchbox.ads.InterstitialAd;
• import com.punchbox.exception.PBException;
• import com.punchbox.listener.AdListener;
•
import android.app.Activity;
import android.os.Bundle;
import android.view.View;

public class InterstitialActivity extends Activity implements AdListener{

    InterstitialAd ad;
    String placementID = "你的广告位ID";//没有可以传空

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_interstitial);

        //声明MoreGameAd实例
        ad = new InterstitialAd (this);
        //设置监听，本类继承了AdListener
        ad.setAdListener(this);

```

```
//设置关闭模式，默认是关闭按钮模式，如果想设成倒计时，调用下面的代码
//ad.setCloseMode(2);//默认是1，关闭按钮模式，值2为倒计时模式
//设置倒计时时长
//ad.setDisplayTime(15);//设置倒计时时长，如果是默认关闭按钮方式，不需要设置此处
//加载广告
ad.loadAd(new AdRequest());
}
//通过xml布局文件中配置的按钮触发
public void interstitialAd (View v){
    try {
        ad.showFloatView(this, 0.9, placementID);
    } catch (PBException e) {
        //当设置的scale不在范围内，或者广告还未加载完毕
        e.printStackTrace();
    }
}

//以下监听回调
@Override
public void onReceiveAd() {
    // 对于固定广告在收到广告后回调，对于有预加载的弹出和精品推荐广告，仅仅是在预加载成功后回调，但不一定会展示
}

@Override
public void onFailedToReceiveAd(PBException ex) {
    // 在请求广告失败后回调
}

@Override
public void onPresentScreen() {
    // 展示广告时回调
}

@Override
public void onDismissScreen() {
    //广告移除时回调
}
}
```

**提示:**

为了能更快地展示弹出广告，减少因网络原因带来的延迟，建议将广告实例化和加载部分放在整个应用启动部分，整个应用运行过程当中，只需要调用一次 `loadAd()`，并不需要每个页面都调用 `loadAd()`。当广告加载完成，会在回调函数 `onReceivedAd()` 中收到通知，开发者也可以在此函数中启动 `showFloatView()`。

接口说明：

| 接口名   | 功能描述                       | 参数说明   |
|---|----------------------------|--|
| <code>showFloatView(Activity ac,double scale,String placementID)</code> | 以浮动窗的方式在当前 Activity 展示弹出广告 | 1.activity: 要展示弹出广告的 Activity 对象；<br>2.scale: 弹出广告显示视图窗口相对屏幕大小的比例，必须大于 0.5，小于等于 1.0<br>3. <code>placementID</code> : 广告位 ID，参见【嵌入弹出广告】中对此的定义。没有可以传空。 |
| <code>show(ViewGroup parent,String placementID)</code>                  | 给定一个容器，弹出广告将展示在此容器内        | 1.parent: 装载弹出广告的容器，可以是各种布局。<br>2. <code>placementID</code> : 广告位 ID，参见【嵌入弹出广告】中对此的定义。没有可以传空。  |
| <code>setCloseMode(int mode)</code>                                     | 设置关闭模式，                    | mode: 整数，1: 关闭按钮方式；<br>2: 倒计时方式，默认为 1。   |
| <code>setDisplayTime(int seconds)</code>                                | 设置倒计时时长                    | 单位为秒   |
| <code>dismiss()</code>  | 从当前屏幕上移除广告，跟点击关闭按钮效果一致。    |  |
| <code>destroy()</code>  | 销毁                         |  |

## (5)、积分墙广告：

步骤：

- 声明 `OfferWallAd` 实例
- 设置监听(可选)
- 对其加载广告
- 在需要展示的时机展示精品推荐；

```

import com.punchbox.ads.AdRequest;
import com.punchbox.ads.OfferWallAd;
import com.punchbox.exception.PBException;
import com.punchbox.listener.AdListener;
import android.app.Activity;
```



```
import android.os.Bundle;
import android.view.View;

public class OfferWallActivity extends Activity implements AdListener{

    OfferWallAd ad;
    String placementID = "你的广告位ID";//没有可以传空

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_moregame);

        //声明OfferWallAd实例
        ad = new OfferWallAd(this);
        //设置监听，本类继承了AdListener
        ad.setAdListener(this);
        //加载广告
        ad.loadAd(new AdRequest());
    }
    //通过xml布局文件中配置的展示按钮触发
    public void offerWall(View v){
        try {
            ad.showFloatView(this, 0.9, placementID);
        } catch (PBException e) {
            //当设置的scale不在范围内，或者广告还未加载完毕
            e.printStackTrace();
        }
    }

    //以下监听回调
    @Override
    public void onReceiveAd() {
        // 对于固定广告在收到广告后回调，对于有预加载的弹出和精品推荐广告，仅仅是在预加载成功后回调，但不一定会展示
    }

    @Override
    public void onFailedToReceiveAd(PBException ex) {
        // 在请求广告失败后回调
    }
}
```

```

@Override
public void onPresentScreen() {
    // 展示广告时回调
}

@Override
public void onDismissScreen() {
    //广告移除时回调
}
}

```

### 提示:

为了能更快地展示积分墙广告，减少因网络原因带来的延迟，建议将广告实例化和加载部分放在整个应用启动部分。整个应用运行过程当中，只需要调用一次 `loadAd()`，并不需要每个页面都调用 `loadAd()`。当广告加载完成，会在回调函数 `onReceivedAd()` 中收到通知，开发者也可以在此函数中启动 `showFloatView()`。

用户在完成任务后，服务器会返回给开发者已经完成的任務列表，同时也会返回之前已经领取过积分但没有跟服务器进行确认的任务列表。开发者可以根据状态值确定哪些任务是已经完成还没有领取积分，哪些是已经领取积分但未告知服务器。已经领取但未告知服务器的需要开发者调用确认接口进行确认，以免每次查询都会返回。

查询积分返回方式:

(1).通过查询接口进行查询:

```
double coins = ad.queryPoints();//ad 是上面定义的 OfferWallAd 实例
```

(2).通过注册监听获取积分返回:

在定义 `OfferWallAd` 实例后并在调用展现 `showFloatView()` 方法前，可以注册积分查询监听，`sdk` 将在积分墙广告每次展现时查询是否有积分返回，如查询到有积分返回，将通过监听回调。

```

mAd = new OfferWallAd(this);

mAd.setPointsChangeListener(new PointsChangeListener() {

    @Override
    public void onPointsChanged(double coins) {
        //TODO 处理返回的积分
    }
});
AdRequest request = new AdRequest();

```

```
mAd.loadAd(request);
```

接口说明：

| 接口名  | 功能描述                       | 参数说明   |
|--|----------------------------|--|
| showFloatView(Activity ac,double scale,String placementID) | 以浮动窗的方式在当前 Activity 展示精品推荐 | <p>1.activity :要展示精品推荐的 Activity 对象；</p> <p>2.scale: 积分墙显示窗口相对屏幕大小的比例，必须大于 0.5，小于等于 1.0</p> <p>3. placementID:广告位 ID，参见【嵌入积分墙广告】中对此的定义。没有可以传空。</p> |
| show(ViewGroup parent,String placementID)                  | 给定一个容器，积分墙广告将展示在此容器内       | <p>1.parent: 装载积分墙的容器，可以是各种布局。</p> <p>2. placementID:广告位 ID，参见【嵌入积分墙广告】中对此的定义。没有可以传空。</p>  |
| queryPoints()  | 查询是否有积分返回                  | 返回为 0 或具体积分  |
| setPointsChangeListener(PointsChangeListener listener)     | 设置积分返回监听                   | 有积分返回时返回积分，会在每次展示时查询，如果有则回调。   |
| dismiss()  | 从当前屏幕上移除广告，跟点击关闭按钮效果一致。    |  |
| destroy()  | 销毁                         |  |

## （6）、积分墙按钮：

Punchbox SDK 还提供一种一键封装积分墙广告的按钮，开发者只需在应用中嵌入该按钮，无需编写加载广告代码即可实现打开积分墙。

在 xml 布局中，嵌入 OfferWallButton，如下：

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity" >

    <com.punchbox.ads.OfferWallButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="您需要替换的图片资源，如果不需要替换，将采用默认的图片，请删除本行"
        ads:placementID="请替换成您的广告位id，如果没有，去掉此行"/>

</LinearLayout>
```

无需编写任何代码，直接封装了打开积分墙广告功能,同时支持在 xml 中设置按钮图片(xml 设置图片可以使用:android:src=""属性指定图片,如果采用默认图片，删除该行)。

开发者也可以在代码中通过实例化积分墙 Button 对象来创建按钮，支持设置按钮图片和积分墙展示 scale 值。

接口说明：

| 接口名   | 功能描述                                      | 参数说明                              |
|---|---|-----------------------------------|
| setImageBitmap(Bitmap bm)                     | 设置按钮图片                                    | bm:Bitmap 类型，需要设置为按钮的图片           |
| setImageDrawable(Drawable drawable)           | 设置按钮图片                                    | drawable: Drawable 类型，需要设置为按钮的图片  |
| setImageResource(int resId)                   | 设置按钮图片资源                                  | resId:int 类型，需要设置为按钮图片的资源 id      |
| setAdListener(AdListener adListener)          | 设置广告监听                                    | adListener:AdListener 类型          |
| setScale(float scale)                         | 设置点击按钮后，弹出的积分墙视图相对于屏幕大小，如果 scale 为 1，将填满。 | scale: float 类型，必须大于 0.5，小于等于 1.0 |
| queryPoints()                                 | 查询是否有积分返回                                 | 返回为 0 或积分                         |
| setPointsChangeListener(PointsChangeListener) | 设置积分返回监听                                  | 如果有设置此监听，则会在每                     |

|           |  |                   |
|-----------|--|-------------------|
| listener) |  | 次积分墙展示时查询是否有积分返回。 |
|-----------|--|-------------------|

## 5. 常见错误码说明:

| 错误码  | 说明   |
|------|--|
| 1000 | 服务未开启  |
| 1001 | http 方法错误  |
| 1002 | 参数错误   |
| 1003 | POST 数据错误  |
| 1004 | 事件关闭   |
| 1005 | punchbox 关闭  |
| 1006 | 无广告适合的广告组  |
| 1007 | 无广告创意可用  |
| 1008 | 创意 URL 不存在   |
| 1009 | 广告模板不存在  |
| 1010 | App 关闭   |
| 1011 | App 版本关闭   |
| 1012 | 未支持的广告类型   |
| 1013 | 频繁的请求  |
| 1014 | 地域开关关闭   |
| 1015 | 作弊请求   |
| 2000 | 手机没有网络连接   |
| 2001 | 网络错误, 如超时或无法连接到服务器                                 |
| 2002 | 服务器返回内容为空  |
| 2003 | Android SDK 版本太低, 广告将不做响应, 建议 Android SDK 版本最低 2.2 |
| 2004 | http 错误(http 状态码非 200)                             |
| 2005 | 广告开关关闭或者没有调用 loadAd 方法                             |
| 2006 | 广告还未加载完成   |
| 2007 | 广告显示比例相对于屏幕的比例不在范围内, 必须大于 0.5, 小于等于 1.0            |

|      |                              |
|------|------------------------------|
| 2008 | 传递的 Context 对象不是 Activity 类型 |
| 2009 | Context 对象已被销毁               |
| 2010 | 广告素材加载超时                     |
| 9999 | 未知错误                         |

## 6.注意事项:

### (1). 广告叠加注意事项:

请开发者在同一个页面同一时间不要同时展示多种广告, 如展示固定广告的同时, 又展示全屏广告, 导致全屏广告遮盖固定广告, 造成一些无用的数据。可以参考 Demo 中的 `BannerWithMoreGameActivity` 类;

### (2). 混淆注意事项:

如果您的应用是通过混淆再进行发布, 请不要混淆 SDK 中的内容(SDK 已经是经过混淆和压缩), 以免影响正常使用, 请在混淆配置中增加下面的内容进行保留。

```
-keepattributes Exceptions
-keepattribute Signature
-keepattribute Deprecated
-keepattributes *Annotation*
-keep class * extends com.punchbox.ads.Ad {
    public *;
}

-keep class com.punchbox.ads.AdRequest {
    public *;
}

-keep class com.punchbox.ads.MoreGameButton {
    public *;
}

-keep class com.punchbox.response.TaskInfo {
    public *;
}

-keep class com.punchbox.ads.OfferWallButton {
    public *;
}

-keep class com.punchbox.exception.PBException {
```

```
    public *;
}

-keep class com.punchbox.listener.AdListener {
    public *;
}

-keep class com.punchbox.listener.PointsChangeListener {
    public *;
}

-keep class * extends android.app.Service {
    public *;
}

-keepclassmembers class com.punchbox.report.ReportData {
    public *;
}

-keepclassmembers class com.punchbox.engine.DownloadData {
    public *;
}

-keepclassmembers class com.punchbox.engine.DownloadSubData {
    public *;
}

-keepclassmembers class com.punchbox.data.DownloadCompleteInfo {
    public *;
}

-keepclassmembers class com.punchbox.data.AppInfo {
    public *;
}
```