Domob Android SDK 安装手册

目录

1. SDK 的嵌入

欢迎您使用 Domob Android SDK 3.0 系列。本着更方便开发者嵌入和使用 SDK 的原则，Domob Android SDK 3.0 版本在集成和使用方式上都有较大的变动，因此无论您是否是第一次使用 Domob SDK，都建议您仔细的阅读以下内容。为了能够正常接收 Domob 的广告，请您严格按照以下步骤嵌入 SDK。文档中的示例代码段都可以在 Sample 工程中找到。

特别注意：一个应用中只能使用唯一的 PublisherID，并且保证这个 ID 是您在多盟官网上和应用包名绑定过的。一个应用中可以创建多个类型的广告位，并且获得对应这个广告位的唯一 PlacementID。在创建多盟广告视图的时候，需要填入这两个 ID。

注意：

从 3.3.0 版本我们增加了广告位功能，SDK 所有请求广告的接口参数都增加了广告位 placementID，您需要去多盟官网手动生成 placementID。

从 3.3.5 开始，inline 添加新类型 Flexible banner，它的特性是宽度可以自适应屏幕，高度在手机和平板上分别是 50、90（逻辑像素），提升了广告的展现效果。

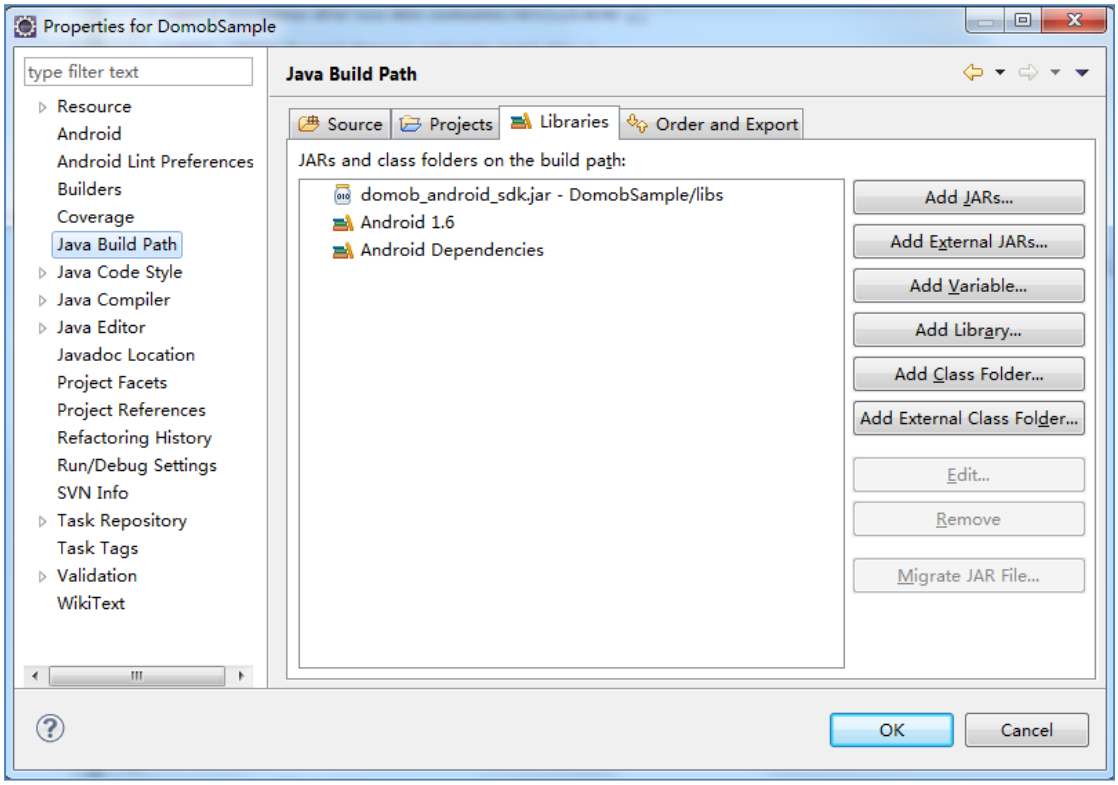
从 4.0.0 开始，新增 feeds 广告类型，具体请参考文档中相关目录或通过 sample 查看效果。

步骤 1：添加 Domob Android SDK 到工程中

首先请在您的工程文件根目录下创建一个名为 libs 的子目录，并将 Domob SDK 的 JAR 包(domob_android_sdk.jar)拷贝到 libs 目录下。

对于 Eclipse 工程，请参照下面的步骤添加 JAR 包：

1) 在 “Package Explorer” 页签中右击你的工程并选择 “Properties”
2) 在左侧面板中选择 “Java Build Path”
3) 在主窗口中选择 “Libraries”页签
4) 点击“Add JARs…”按钮
5) 选择您拷贝到 libs 目录下的 domob_android_sdk.jar
6) 点击 “OK” 完成添加，如图所示：



步骤 2：修改 AndroidManifest.xml 文件

1. 添加 Activity 声明

在 AndroidManifest.xml 文件中的</application>标签之前，还需要添加必要的 Activity 声明，添加如下代码：

```
<activity android:name="cn.domob.android.ads.DomobActivity"
    android:theme="@android:style/Theme.Translucent"></activity>
```

2. 添加权限许可

在 AndroidManifest.xml 文件中的<application>标签之前，请为 SDK 添加以下权限许可（如果 App 本身没有的话）：

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

一些需要精准位置，当您完成了以上的步骤设置后，请参考以下完整的 AndroidManifest.xml 范例：

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cn.domob.ads.sample" android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="4" />

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

    <application android:icon="@drawable/ic_launcher"
        android:label="@string/app_name">
        <activity android:name=".DomobSampleActivity" android:label="@string/app_name">
        </activity>
        <activity android:name="cn.domob.android.ads.DomobActivity"
            android:theme="@android:style/Theme.Translucent"></activity>
        <activity android:name="BannerAd"></activity>
        <activity android:name="FlexibleBannerAd" android:screenOrientation="portrait"></activity>
        <activity android:name="RotatableFlexibleBannerAd"  android:configChanges="orientation"></activity>
        <activity android:name="InterstitialAd"></activity>
        <activity android:name="FeedsAd"></activity>
        <activity android:name="SplashScreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>
</manifest>
```

2. 广告视图的使用

当您完成了以上步骤，您就可以正常的使用 Domob 的广告视图控件来请求和显示广告了。Domob SDK 为您提供了多种广告展现形式，丰富广告的展现并且提升广告位的效果。

inline 添加新类型 Flexible banner，它的特性是宽度自适应屏幕，高度在手机和平板上分别是 50、90（逻辑像素）.

具体广告类型如下表所示：

广告类型：inline		
Dimension（逻辑像素）	Density（密度）	Device（适用设备）
320x50	1/1.5/2	android 手机
728x90	1	android 平板
600x94	1	android 平板
300x250	1/1.5/2	android 手机 android 平板
600x500	1	android 平板

广告类型：插屏广告		
Dimension（逻辑像素）	Density（密度）	Device（适用设备）

300x250	1/1.5/2	android 手机
600x500	1	android 平板
full screen（横竖屏）	-	android 手机 android 平板

广告类型：开屏广告		
Dimension（逻辑像素）	Density（密度）	Device（适用设备）
full screen（横竖屏）	-	iPhone android 手机 iPad android 平板

Banner 类型广告视图的添加

添加普通 banner：

由于简化了开发者嵌入 SDK 的步骤,因此无法以在 layout 的 xml 文件中创建 view 的方式来创建广告视图。如果您需要通过 layout 文件来定位广告位，可以在想要加广告视图的位置预留一个 ViewGroup(例如 RelativeLayout)来作为 AD container。如下示例：

```java
public class BannerAd extends Activity {
    RelativeLayout mAdContainer;
    DomobAdView mAdview320x50;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.banner);

        mAdContainer = (RelativeLayout) findViewById(R.id.adcontainer);
        //创建一个320x50的广告View
        mAdview320x50 = new DomobAdView(this, DomobSampleActivity.PUBLISHER_ID,
                DomobSampleActivity.InlinePPID, DomobAdView.INLINE_SIZE_320X50);
        mAdview320x50.setKeyword("game");
        mAdview320x50.setUserGender("male");
        mAdview320x50.setUserBirthdayStr("2000-08-08");
        mAdview320x50.setUserPostcode("123456");

        //设置广告view的监听器。
        mAdview320x50.setAdEventListener(new DomobAdEventListener() {

            @Override
            //成功接收到广告返回回调
            public void onDomobAdReturned(DomobAdView adView) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "onDomobAdReturned");
            }

            @Override
            //Landing Page成功打开回调
            public void onDomobAdOverlayPresented(DomobAdView adView) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "overlayPresented");
            }

            @Override
            //Landing Page关闭回调
            public void onDomobAdOverlayDismissed(DomobAdView adView) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "Overrided be dismissed");
            }

            @Override
            //广告点击回调
            public void onDomobAdClicked(DomobAdView arg0) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "onDomobAdClicked");
            }

            @Override
            //广告请求失败回调
            public void onDomobAdFailed(DomobAdView arg0, ErrorCode arg1) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "onDomobAdFailed");
            }

            @Override
            //离开应用回调
            public void onDomobLeaveApplication(DomobAdView arg0) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "onDomobLeaveApplication");
            }

            @Override
            //返回当前的Context
            public Context onDomobAdRequiresCurrentContext() {
                // TODO Auto-generated method stub
                return BannerAd.this;
            }
        });
        //将广告View增加到视图中。
        mAdContainer.addView(mAdview320x50);
    }
}
```

至此广告视图就成功的添加到 App 中，当 App 中放有广告视图的界面显示后，广告视图会自动请求广告，成功获取广告后会自动出现在界面上。当该界面隐藏或销毁时，广告视图也随之停止请求和刷新，并随之销毁。

创建 Banner 广告位，请在官网“广告位管理”界面的“广告位类型设置中”请选择“inline 广告位”。

添加 Flexible banner：

inline 添加新类型 Flexible banner，它的特性是宽度自适应屏幕，高度在手机和平板上分别是 50、90（逻辑像素）。

（1）如果应用界面不随屏幕旋转，那么它的用法和普通 banner 相同，只需要在 DomobAdView 构造函数中传入类型 DomobAdView.INLINE_SIZE_FLEXIBLE。

（2）如果应用界面跟随屏幕旋转，有以下两种实现方式

方式 1：

首先在 AndroidManifest.xml 承载 DomobAdView 的 Activity 中添加 configChanges

```
<activity android:name="RotatableFlexibleBannerAd"  android:configChanges="orientation"></activity>
```

然后在承载 DomobAdView 的 Activity 类中重写 onConfigurationChanged，并且在实现方法中调用 DomobAdView 对象方法 orientationChanged ()。也许由于应用自身需求，configChanges 不只包含 orientation，从而造成误调 orientationChanged ()，针对这种情况 SDK 内部做了相关处理，可以过滤掉不是因为屏幕旋转引起的调用。

```
@Override
public void onConfigurationChanged(Configuration configuration) {
    super.onConfigurationChanged(configuration);
    ((DomobAdView) mAdContainer.getChildAt(0)).orientationChanged();
}
```

方式 2：

每次旋转都重新生成 DomobAdView 对象

这两种方式同样也都需要在 DomobAdView 构造函数中传入类型 DomobAdView.INLINE_SIZE_FLEXIBLE。

创建 Flexible 广告位，请在官网“广告位管理”界面的“广告位类型设置中”请选择“inline 广告位”

具体用法如下示例：

```
public class FlexibleBannerAd extends Activity {
    RelativeLayout mAdContainer;
    DomobAdView mAdviewFlexibleAdView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.banner);

        mAdContainer = (RelativeLayout) findViewById(R.id.adcontainer);
        // Create ad view
        mAdviewFlexibleAdView = new DomobAdView(this, DomobSampleActivity.PUBLISHER_ID,
                DomobSampleActivity.FlexibleInlinePPID, DomobAdView.INLINE_SIZE_FLEXIBLE);
        mAdviewFlexibleAdView.setKeyword("game");
        mAdviewFlexibleAdView.setUserGender("male");
        mAdviewFlexibleAdView.setUserBirthdayStr("2000-08-08");
        mAdviewFlexibleAdView.setUserPostcode("123456");

        mAdviewFlexibleAdView.setAdEventListener(new DomobAdEventListener() {

            @Override
            public void onDomobAdReturned(DomobAdView adView) {
                Log.i("DomobSDKDemo", "onDomobAdReturned");
            }

            @Override
            public void onDomobAdOverlayPresented(DomobAdView adView) {
                Log.i("DomobSDKDemo", "overlayPresented");
            }

            @Override
            public void onDomobAdOverlayDismissed(DomobAdView adView) {
                Log.i("DomobSDKDemo", "Overrided be dismissed");
            }

            @Override
            public void onDomobAdClicked(DomobAdView arg0) {
                Log.i("DomobSDKDemo", "onDomobAdClicked");
            }

            @Override
            public void onDomobAdFailed(DomobAdView arg0, ErrorCode arg1) {
                Log.i("DomobSDKDemo", "onDomobAdFailed");
            }

            @Override
            public void onDomobLeaveApplication(DomobAdView arg0) {
                Log.i("DomobSDKDemo", "onDomobLeaveApplication");
            }

            @Override
            public Context onDomobAdRequiresCurrentContext() {
                return FlexibleBannerAd.this;
            }
        });

        mAdContainer.addView(mAdviewFlexibleAdView);
    }
}
```

插屏类型广告视图的显示

注意：插屏和 Banner 广告的使用是不同的，插屏广告在使用的时候要先调用 loadInterstitialAd 方法请求到广告，在合适的时机调用 showInterstitialAd 方法来展示插屏广告；在展示结束时通过 loadInterstitialAd 方法请求一次广告，便于在下次使用

时直接通过 showInterstitialAd 方法来展示广告，减少一次广告请求时间。

插屏属于弹出式显示广告的形式。Domob SDK 提供给您检查当前是否可弹出的监听以及弹出显示广告的接口。如下示例：

```
public class InterstitialAd extends Activity {
    DomobInterstitialAd mInterstitialAd;
    Button mInterstitialBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.interstital);

        mInterstitialAd = new DomobInterstitialAd(this, DomobSampleActivity.PUBLISHER_ID,
                DomobSampleActivity.InterstitialPPID, DomobInterstitialAd.INTERSITIAL_SIZE_300X250);

        mInterstitialBtn = (Button) findViewById(R.id.interstitial);

        mInterstitialAd.setInterstitialAdListener(new DomobInterstitialAdListener() {
            @Override
            public void onInterstitialAdReady() {
                Log.i("DomobSDKDemo", "onAdReady");
            }

            @Override
            public void onLandingPageOpen() {
                Log.i("DomobSDKDemo", "onLandingPageOpen");
            }

            @Override
            public void onLandingPageClose() {
                Log.i("DomobSDKDemo", "onLandingPageClose");
            }

            @Override
            public void onInterstitialAdPresent() {
                Log.i("DomobSDKDemo", "onInterstitialAdPresent");
            }

            @Override
            public void onInterstitialAdDismiss() {
                // Request new ad when the previous interstitial ad was closed.
                mInterstitialAd.loadInterstitialAd();
                Log.i("DomobSDKDemo", "onInterstitialAdDismiss");
            }

            @Override
            public void onInterstitialAdFailed(ErrorCode arg0) {
                Log.i("DomobSDKDemo", "onInterstitialAdFailed");
            }
```

```
            @Override
            public void onInterstitialAdLeaveApplication() {
                Log.i("DomobSDKDemo", "onInterstitialAdLeaveApplication");

            }

            @Override
            public void onInterstitialAdClicked(DomobInterstitialAd arg0) {
                Log.i("DomobSDKDemo", "onInterstitialAdClicked");
            }
        });

        mInterstitialAd.loadInterstitialAd();
        mInterstitialBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                if (mInterstitialAd.isInterstitialAdReady()){
                    mInterstitialAd.showInterstitialAd(InterstitialAd.this);
                } else {
                    Log.i("DomobSDKDemo", "Interstitial Ad is not ready");
                    mInterstitialAd.loadInterstitialAd();
                }
            }
        });
    }
}
```

开屏广告视图的显示

开屏广告的使用场景是在应用刚刚开启时，在 SDK 内分为缓存开屏和实时开屏两种。缓存开屏的加载机制为，本次加载，缓存本次的广告应答以及相关资源，下次开启时展现。而实时开屏在每次开启时都会实时请求广告并加载和缓存资源，在超时时间前加载完成即自动展现。详细使用方法可见 Sample 中的 SplashScreen 类中使用的示例代码。

	广告请求流程	优点	缺点
缓存开屏	当次缓存，下次展现	无需等待广告 loading，体验好。	填充至展现流失率较高
实时开屏	当次加载，当次展现，超时放弃（超时时限开发者可设）	填充至展现流失率较低。	用户需要等待

使用注意事项：

1) 使用开屏广告前要确保您的应用有自己的开机画面，并且在开屏的界面上提供一个 Layout 作为开屏广告的容器。
2) 在开屏广告被关闭或展现完成需要自动关闭时，会回调 onSplashDismiss/onRTSplashDismiss 方法，通知开发者开屏被关闭。在这个方法中，强烈建议开发者进行开屏页面到主页面的跳转（即 Activity 跳转），这样给用户带来的是开屏到主屏的无缝切换体验。
3) 建议开发者在开屏界面的 Activity 中，通过 onKeyDown 方法，屏蔽开屏界面的返回键行为（KeyEvent.KEYCODE_BACK），以此来保证您的开屏和多盟的开屏广告正常显示完成。

开屏属于弹出式显示广告的形式。SDK 包中的 Sample 提供了完整的开屏使用流程，建议开发者查看并在自己的应用/游戏中实现该使用方法。

feeds 类型广告视图的显示

feeds 广告类型适合具有列表下拉刷新场景的应用添加，当用户为了更新内容，将列表下拉，用户手势松开的时机，在列表的顶部展示广告。

feeds 类型广告使用步骤如下：

- 请将自定义的 ListView 放入 LinearLayout 中，将 LinearLayout 放回 ListView 原有位置。

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/adcontainer"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <cn.domob.ads.sample.MyListView
        android:id="@+id/listView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

- feeds 广告在使用前要先调用 loadFeedsAd 方法请求到广告，推荐在创建 FeedsAd 对象后立即调用。

```
mFeedsAdView = new DomobFeedsAdView(this,
        DomobSampleActivity.PUBLISHER_ID, DomobSampleActivity.FeedsPPID);
mFeedsAdView.loadFeedsAd();
```

- 然后通过调用 LinearLayout 对象方法 addView(mFeedsAdView, 0)的方式将 FeedsAdView 对象插入 LinearLayout 最前端。

```
LinearLayout parentLinearLayout = (LinearLayout) findViewById(R.id.adcontainer);
parentLinearLayout.addView(mFeedsAdView, 0);
```

- 当用户刷新列表内容时调用 showFeedsAd 方法来展示广告。

```
listView.setonRefreshListener(new OnRefreshListener() {
    public void onRefresh() {
        if (mFeedsAdView.isFeedsAdReady()) {
            mFeedsAdView.showFeedsAd(FeedsAd.this);
        } else {
            mFeedsAdView.loadFeedsAd();
        }
```

- 在展示结束时通过 loadFeedsAd 方法请求下一次广告，便于在下次使用时直接通过 showFeedsAd 方法来展示广告，减少一次广告请求时间。

```
@Override
public void onFeedsAdDismiss() {
    Log.i("DomobSDKDemo", "onFeedsAdDismiss");
    mFeedsAdView.loadFeedsAd();
}
```

注意：广告在展现期间被其它 view 遮挡时，请调用 SDK 提供的接口 closeFeedsAd()关闭广告，广告声音会随之消失。

测试多盟广告

在应用通过审核前，对应的正式 Publisher ID 只能展示测试广告。当应用通过审核后，对应的正式 Publisher ID 才能展示正常广告。任何时候，您都可以使用多盟公共测试 Publisher ID 来测试多盟的广告效果及兼容性。使用公共测试 Publisher ID 的应用在任何时候都只展示测试广告。

Android 公共测试 Publisher ID：56OJyM1ouMGoaSnvCK

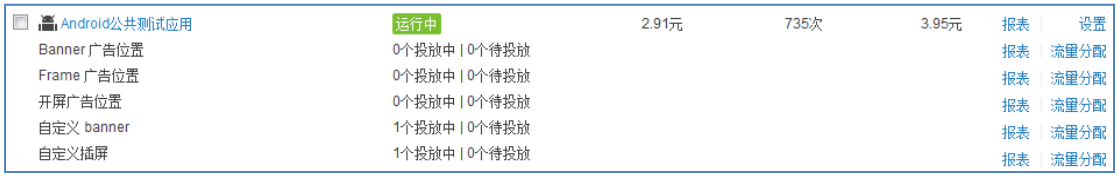
3. 广告位设置与管理

创建广告位

首先，还是要先创建一个应用，第二步，创建广告位并选择广告位类型，确认后会生成一个广告位 ID，然后就可以去配置 SDK 了；注意配置 SDK 时广告位类型一定要和创建时对应上。

广告位管理

登陆多盟官方网站，点击“我的应用”就能管理您应用下面的广告位，点击“流量分配”管理每个广告位展示广告比例的分配。如下图所示：



当您成功的完成以上步骤，您的应用已经可以正常的请求广告了。

本地管理

Domob SDK 提供了更多的接口，通过您的设置，可以增加广告投放的精准程度，从而提高您的广告收入。Domob SDK 还提供了供您监听广告视图工作的接口，帮助您更好的了解和

调整广告视图与 App 的工作配合。如下示例：

```
        //创建一个320x50的广告View
        mAdview320x50 = new DomobAdView(this, DomobSampleActivity.PUBLISHER_ID,
                DomobSampleActivity.InlinePPID, DomobAdView.INLINE_SIZE_320X50);
        mAdview320x50.setKeyword("game");
        mAdview320x50.setUserGender("male");
        mAdview320x50.setUserBirthdayStr("2000-08-08");
        mAdview320x50.setUserPostcode("123456");

        //设置广告view的监听器。
        mAdview320x50.setAdEventListener(new DomobAdEventListener() {

            @Override
            //成功接收到广告返回回调
            public void onDomobAdReturned(DomobAdView adView) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "onDomobAdReturned");
            }

            @Override
            //Landing Page成功打开回调
            public void onDomobAdOverlayPresented(DomobAdView adView) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "overlayPresented");
            }

            @Override
            //Landing Page关闭回调
            public void onDomobAdOverlayDismissed(DomobAdView adView) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "Overrided be dismissed");
            }

            @Override
            //广告点击回调
            public void onDomobAdClicked(DomobAdView arg0) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "onDomobAdClicked");
            }

            @Override
            //广告请求失败回调
            public void onDomobAdFailed(DomobAdView arg0, ErrorCode arg1) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "onDomobAdFailed");
            }

            @Override
            //离开应用回调
            public void onDomobLeaveApplication(DomobAdView arg0) {
                // TODO Auto-generated method stub
                Log.i("DomobSDKDemo", "onDomobLeaveApplication");
            }

            @Override
            //返回当前的Context
            public Context onDomobAdRequiresCurrentContext() {
                // TODO Auto-generated method stub
                return BannerAd.this;
            }
        });
        //将广告View增加到视图中。
        mAdContainer.addView(mAdview320x50);
    }
}
```

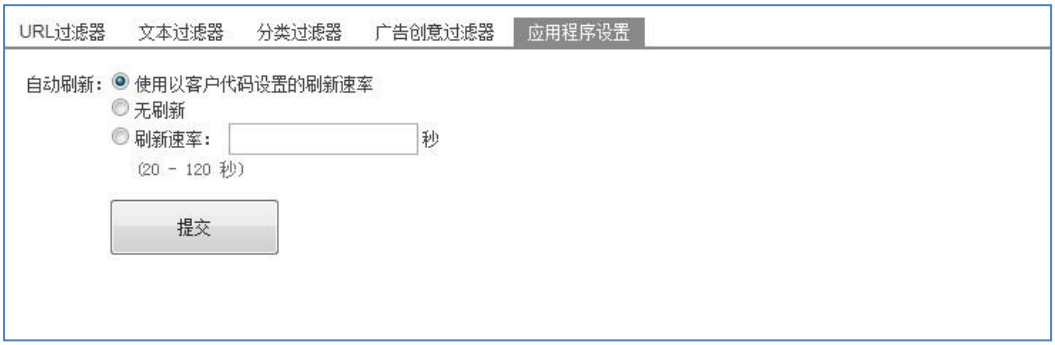
云端管理

除了在代码中，您还可以通过多盟官网管理您的应用。例如，如果需要暂停/开始您的应用，请按照如下路径：

1） 登陆多盟官方网站，点击“我的应用”；

2） 选择相应的应用，并点击“暂停”或者“运行”。

您还可以通过在多盟的官网中设置广告刷新时间，如下图所示：



4. 使用更新提醒功能

要使用 SDK 提供的更新提醒功能，您需要完成以下步骤：

1) 在合适的位置（如应用开启时）调用如下代码：
```
DomobUpdater.checkUpdate(this, "Your Publisher ID.");
```

2) 登录 domob 官网，上传您使用了更新提醒功能的 APK。

3) 当您的应用审核通过后，您就可以在“我的应用” -> “应用列表”（http://www.domob.cn/publisher/app/list）中使用“让多盟帮您推送更新”功能，来帮助您更新应用。

5. 注意事项

在使用 domob SDK 的时候一定要注意以下问题：

1) 一个应用中只能使用唯一的 PublisherID，并且保证这个 ID 是您在多盟官网上和应用包名绑定过的。

2) 插屏和 Banner 广告的使用是不同的，插屏广告在使用的时候要先调用 `loadInterstitialAd` 方法请求到广告，在合适的时机调用 `showInterstitialAd` 方法来展示插屏广告。

3) 插屏广告在展示结束时通过 `loadInterstitialAd` 方法请求一次广告，便于在下次使用时直接通过 `showInterstitialAd` 方法来展示广告，减少一次广告请求的时间。

4) 使用开屏广告前要确保您的应用有自己的开机画面，否则有可能开屏广告在展现前会有 1-2 秒的黑屏出现。这是因为开屏广告从加载资源到广告展示需要 1-2 秒的时间来完成。