

目录

目录	1
Alx Unity插件对接文档	2
Alx Unity插件	2
导入Alx Unity插件	2
版本升级	2
Unity初始化Alx SDK	2
构建AlxAdManager 对象:	2
构建方法 :	2
初始化SDK:	2
AlxAdManager接口说明 :	2
初始化回调接口说明 :	3
初始化实现案例 :	3
Banner广告	3
简介	3
使用说明 : Android	3
AlxBannerAD接口说明 :	4
Banner监听回调说明 :	4
Banner实现案例 :	4
插屏广告	5
简介	5
使用说明 : Android	5
AlxInterstitialAD接口说明 :	5
插屏监听回调说明 :	6
插屏实现案例 :	6
激励视频广告	7
简介	7
使用说明 : Android	7
AlxRewardAD接口说明 :	7
激励视频监听回调说明 :	8
激励视频实现案例 :	8

# Alx Unity插件对接文档

## Alx Unity插件

### 导入Alx Unity插件

借助Alx Unity 插件，Unity 开发者无需编写 Java代码，即可轻松地在 Android应用上展示广告。

在 Unity 编辑器中打开您的项目，然后依次选择 Assets > Import Package > Custom Package，并找到您下载的AlxSDK.unitypackage文件。确保选择所有文件。

### 版本升级

删除原先SDK的aar包，将获取Android端对应的SDK中的aar文件拖拽到Plugins\Android\libs文件夹下即可。

### Unity初始化Alx SDK

构建AlxAdManager 对象:

AlxAdManager作为广告初始化和各种广告加载入口。

构建方法：

```
private AlxAdManager adManager;
private AlxAdManager AdManager
{
    get
    {
        if (adManager == null)
        {
            adManager = AlxSDK.CreateAdManager();
        }
        return adManager;
    }
}
```

初始化SDK:

在初始化前，先申请开通账号，配置好自己的应用，获取Token，SID和APP\_ID。

```
AdManager.Init(ALX_TOKEN, ALX_SID, ALX_APP_ID,new SdkInitCallback());
public class SdkInitCallback : ISdkInitCallback
{
    public void OnInit(bool b, string s)
    {
        // 初始化成功后才能请求广告，否则可能会影响填充率
    }
}
```

AlxAdManager接口说明：

```
public sealed class AlxAdManager
{
    /// <summary>
    /// 无回调初始化
    /// </summary>
    public void Init(string token, string sid, string appid)
    {}
    /// <summary>
```

```

    /// 有回调初始化
    /// </summary>
    public void Init(string token, string sid, string appid, ISdkInitCallback callback)
    {}
    /// <summary>
    /// 加载banner广告
    /// </summary>
    public void LoadBannerAd(string bannerAdPid, IBannerADListener listener, int position, AndroidJavaObject size)
    {}
    /// <summary>
    /// 加载插屏广告
    /// </summary>
    public void LoadInterstitialAd(string alxInterstitialAdPid, IInterstitialADListener listener)
    {}
    /// <summary>
    /// 加载激励视频广告
    /// </summary>
    public void LoadRewardAd(string alxRewardAdPid, IRewardVideoADListener listener)
    {}
}

```

初始化回调接口说明：

```

public interface ISdkInitCallback
{
    //初始化回调
    void OnInit(bool b, string s);
}

```

初始化实现案例：

```

private AlxAdManager adManager;
private AlxAdManager AdManager
{
    get
    {
        if (adManager == null)
        {
            adManager = AlxSDK.CreateAdManager();
        }
        return adManager;
    }
}
private void Start()
{
    //AdManager.Init(ALX_TOKEN,ALX_SID,ALX_APP_ID);
    AdManager.Init(ALX_TOKEN, ALX_SID, ALX_APP_ID, new SdkInitCallback());
}
public class SdkInitCallback : ISdkInitCallback
{
    public void OnInit(bool b, string s)
    {
        Debug.Log("SDK Init : " + b + " msg:" + s);
    }
}

```

## Banner广告

简介

在内容底部或顶部显示的小条形广告。不应将Banner广告放置于文本、图片和应用的其他可点击部分，避免误点。

支持的广告尺寸：320\*50，300\*250，320\*480

使用说明：Android

1. 调用AdManager.LoadBannerAd(ALX\_BANNER\_AD\_PID, new BannerADListener(this), AlxBannerAD.BOTTOM, AlxBannerAD.Size(AlxAdSize.SIZE\_320\_50));加载广告，在监听加载完成时初始化alxBannerAD。
2. 通过alxBannerAD.IsReady()判断是否加载完成，alxBannerAD.ShowAd();展示Banner。
3. 提前调用加载方法进行广告的请求，以便需要触发广告时可以快速展示。

AlxBannerAD接口说明：

```
public class AlxBannerAD
{
    /// <summary>
    /// 展示Banner广告
    /// </summary>
    public void ShowAd()
    {
    }
    /// <summary>
    /// 销毁Banner广告
    /// </summary>
    public void Destory()
    {
    }
    /// <summary>
    /// 设置刷新
    /// </summary>
    public void SetRefresh(int i)
    {
    }
    /// <summary>
    /// 设置Banner广告是否能关闭
    /// </summary>
    public void SetCanClosed(bool b)
    {
    }
    /// <summary>
    /// 获取广告是否加载完成
    /// </summary>
    public bool IsReady()
    {
        return false;
    }
    /// <summary>
    /// 设置广告是否自动刷新
    /// </summary>
    public void SetAutoRefresh(bool b)
    {
    }
    /// <summary>
    /// 设置刷新间隔
    /// </summary>
    public void SetRefreshInterval(int i)
    {
    }
}
```

Banner监听回调说明：

```
public interface IBannerADListener
{
    //加载banner后调用
    void OnAdLoaded(AlxBannerAD alxBannerAD);
    //加载错误后调用
    void OnAdError(int i, string s);
    //点击banner后调用
    void OnAdClicked();
    //展示广告时调用
    void OnAdShow();
    //关闭banner时调用
    void OnAdClose();
}
```

Banner实现案例：

```

public void LoadBannerAd()
{
    AdManager.LoadBannerAd(ALX_BANNER_AD_PID, new BannerADListener(this), BannerAD.BOTTOM, BannerAD.Size(AlxAdSize.SIZE_320_50));
}
public void ShowBanner()
{
    if (alxBannerAD.IsReady())
    {
        alxBannerAD.ShowAd();
    }
}
public class BannerADListener : IBannerADListener
{
    private Example example;
    public BannerADListener(Example example)
    {
        this.example = example;
    }
    public void OnAdClicked()
    { }
    public void OnAdClose()
    { }
    public void OnAdError(int i, string s)
    {
        Debug.Log("onAdError:  " + "code:" + i + "  mgs:" + s);
    }
    public void OnAdLoaded(AlxBannerAD alxBannerAD)
    {
        example.alxBannerAD = alxBannerAD;
    }
    public void OnAdShow()
    { }
}

```

## 插屏广告

### 简介

该广告的效果播放全屏的视频，视频一定时间后可跳过，无需全程观看完。

支持的广告尺寸：全屏

### 使用说明：Android

1. 调用AdManager.LoadInterstitialAd(ALX\_INTERSTITIAL\_AD\_PID, new InstertitalADListener(this));加载广告，在监听加载完成时初始化alxInterstitialAD。
2. 通过alxInterstitialAD.IsReady()判断是否加载完成，alxInterstitialAD.ShowAd();展示插屏。
3. 提前调用加载方法进行广告的请求，以便需要触发广告时可以快速展示。
4. 在onInterstitialAdClose回调中，调用加载方法进行广告的请求，方便下一次广告的展示。
5. 禁止在onInterstitialAdLoadFail回调中执行广告加载的方法，否则会引起很多无用请求且可能会导致应用卡顿。

### AlxInterstitialAD接口说明：

```

public class AlxInterstitialAD
{
    /// <summary>
    /// 展示Interstitial广告
    /// </summary>
    public void Show()
    {}
    /// <summary>
    /// 获取广告是否加载完成
    /// </summary>
    public bool IsReady()
    {

```

```

        return false;
    }
    /// <summary>
    /// 销毁广告
    /// </summary>
    public void Destroy()
    {}
    /// <summary>
    /// 设置超时
    /// </summary>
    public void SetTimeout(int i, int i1)
    {}
}

```

插屏监听回调说明：

```

public interface IInterstitialADListener
{
    //加载完插屏调用
    void OnInterstitialAdLoaded(AlxInterstitialAD instertitial);
    //加载失败后调用
    void OnInterstitialAdLoadFail(int i, string s);
    //点击插屏后调用
    void OnInterstitialAdClicked();
    //展示插屏时调用
    void OnInterstitialAdShow();
    //关闭插屏时调用
    void OnInterstitialAdClose();
    //视频开始时调用
    void OnInterstitialAdVideoStart();
    //视频结束时调用
    void OnInterstitialAdVideoEnd();
    //播放失败时调用
    void OnInterstitialAdVideoError(int i, string s);
}

```

插屏实现案例：

```

public void LoadInstertitialAd()
{
    AdManager.LoadInterstitialAd(ALX_INTERSTITIAL_AD_PID, new InstertitalADListener(this));
}
public void ShowInstertitialAd()
{
    if (alxInstertitialAD.IsReady())
    {
        alxInstertitialAD.Show();
    }
}
public class InstertitalADListener : IInterstitialADListener
{
    private Example example;
    public InstertitalADListener(Example example)
    {
        this.example = example;
    }
    public void OnInterstitialAdClicked()
    { }
    public void OnInterstitialAdClose()
    { }
    public void OnInterstitialAdLoaded(AlxInterstitialAD instertitialAD)
    {
        example.alxInstertitialAD = instertitialAD;
    }
    public void OnInterstitialAdLoadFail(int i, string s)
    {
        Debug.Log("onInterstitialAdLoadFail:  " + "code:" + i + "  mgs:" + s);
    }
}

```

```

}
public void OnInterstitialAdShow()
{ }
public void OnInterstitialAdVideoEnd()
{ }
public void OnInterstitialAdVideoError(int i, string s)
{
    Debug.Log("onInterstitialAdVideoError:  " + "code:" + i + "  mgs:" + s);
}
public void OnInterstitialAdVideoStart()
{ }
}

```

## 激励视频广告

### 简介

本SDK为接入方提供激励视频广告，该广告的效果为观看完毕视频广告，发放奖励给用户。

支持的广告尺寸：全屏

### 使用说明：Android

1. 调用AdManager.LoadRewardAd(ALX\_REWARD\_AD\_PID, new RewardADListener(this));加载广告，在监听加载完成时初始化alxRewardAD。
2. 通过alxRewardAD.IsReady()判断是否加载完成，alxRewardAD.ShowAd();展示激励视频。
3. 请提前调用加载方法进行广告的请求（比如在应用启动就开始加载广告），以便需要触发广告时可以快速展示。
4. 在onRewardedVideoAdClosed回调中，调用加载方法进行广告的请求，方便下一次广告的展示。
5. 禁止在onRewardedVideoAdFailed回调中执行广告加载的方法，否则会引起很多无用请求且可能会导致应用卡顿。

### AlxRewardAD接口说明：

```

public class AlxRewardAD
{
    /// <summary>
    /// 展示激励视频广告
    /// </summary>
    public void ShowVideo()
    { }
    /// <summary>
    /// 获取广告是否加载完成
    /// </summary>
    public bool IsReady()
    {
        return false;
    }
    /// <summary>
    /// 销毁广告
    /// </summary>
    public void Destroy()
    { }
    /// <summary>
    /// 设置继续
    /// </summary>
    public void Resume()
    { }
    /// <summary>
    /// 设置暂停
    /// </summary>
    public void Pause()
    { }
    /// <summary>
    /// 设置超时
    /// </summary>
    public void SetTimeout(int i, int i1)
    { }
}

```

激励视频监听回调说明：

```
public interface IRewardVideoADListener
{
    //加载完成调用
    void OnRewardedVideoAdLoaded(AlxRewardAD reward);
    //加载失败调用
    void OnRewardedVideoAdFailed(int i, string s);
    //开始播放视频回调
    void OnRewardedVideoAdPlayStart();
    //视频播放完成回调
    void OnRewardedVideoAdPlayEnd();
    //视频播放失败回调
    void OnRewardedVideoAdPlayFailed(int i, string s);
    //关闭视频回调
    void OnRewardedVideoAdClosed();
    //点击视频时回调
    void OnRewardedVideoAdPlayClicked();
    //获得奖励回调
    void OnReward();
}
```

激励视频实现案例：

```
public void LoadRewardAd()
{
    AdManager.LoadRewardAd(ALX_REWARD_AD_PID, new RewardADListener(this));
}
public void ShowRewardAd()
{
    if(alxRewardAD.IsReady())
    {
        alxRewardAD.ShowVideo();
    }
}
public class RewardADListener: IRewardVideoADListener
{
    private Example example;
    public RewardADListener(Example example)
    {
        this.example = example;
    }
    public void OnReward()
    {}
    public void OnRewardedVideoAdClosed()
    {}
    public void OnRewardedVideoAdFailed(int i, string s)
    {
        Debug.Log("onRewardedVideoAdFailed:  " + "code:" + i + "  mgs:" + s);
    }
    public void OnRewardedVideoAdLoaded(AlxRewardAD reward)
    {
        example.alxRewardAD = reward;
    }
    public void OnRewardedVideoAdPlayClicked()
    {}
    public void OnRewardedVideoAdPlayEnd()
    {}
    public void OnRewardedVideoAdPlayFailed(int i, string s)
    {
        Debug.Log("onRewardedVideoAdPlayFailed:  " + "code:" + i + "  mgs:" + s);
    }
    public void OnRewardedVideoAdPlayStart()
    {}
}
```

