

unity Ads 3.x

基本操作与集成手册

2019.6

内容预览

● Unity后台操作

- Unity ID *4
- Operate后台 *5
- Game ID *6
 - 创建Project
 - 获取Game ID
- Placement ID *9
 - 获取Placement ID
 - 添加Placement
 - 广告位设置
 - 自动发送收入报表

● 广告代码集成

- Asset Store Package。 *15
 - 导入广告资源包
 - 切换平台
 - 广告展示、回调
- iOS SDK *19
 - 导入广告SDK
 - 广告初始化
 - 广告展示
 - 广告回调
- Android SDK *24
 - 导入广告SDK
 - 实现接口
 - 广告初始化
 - 广告展示
 - 广告回调

● 附录

- 填写Store ID *32
- Test Mode *34
- 获得技术支持 *35
- 资源链接 *36

Unity后台操作



- Unity ID
- Operate后台
- Game ID
 - 创建Project
 - 获取Game ID
- Placement ID
 - 获取Placement ID
 - 添加Placement
 - 广告位设置
 - 自动发送收入报表

Unity ID

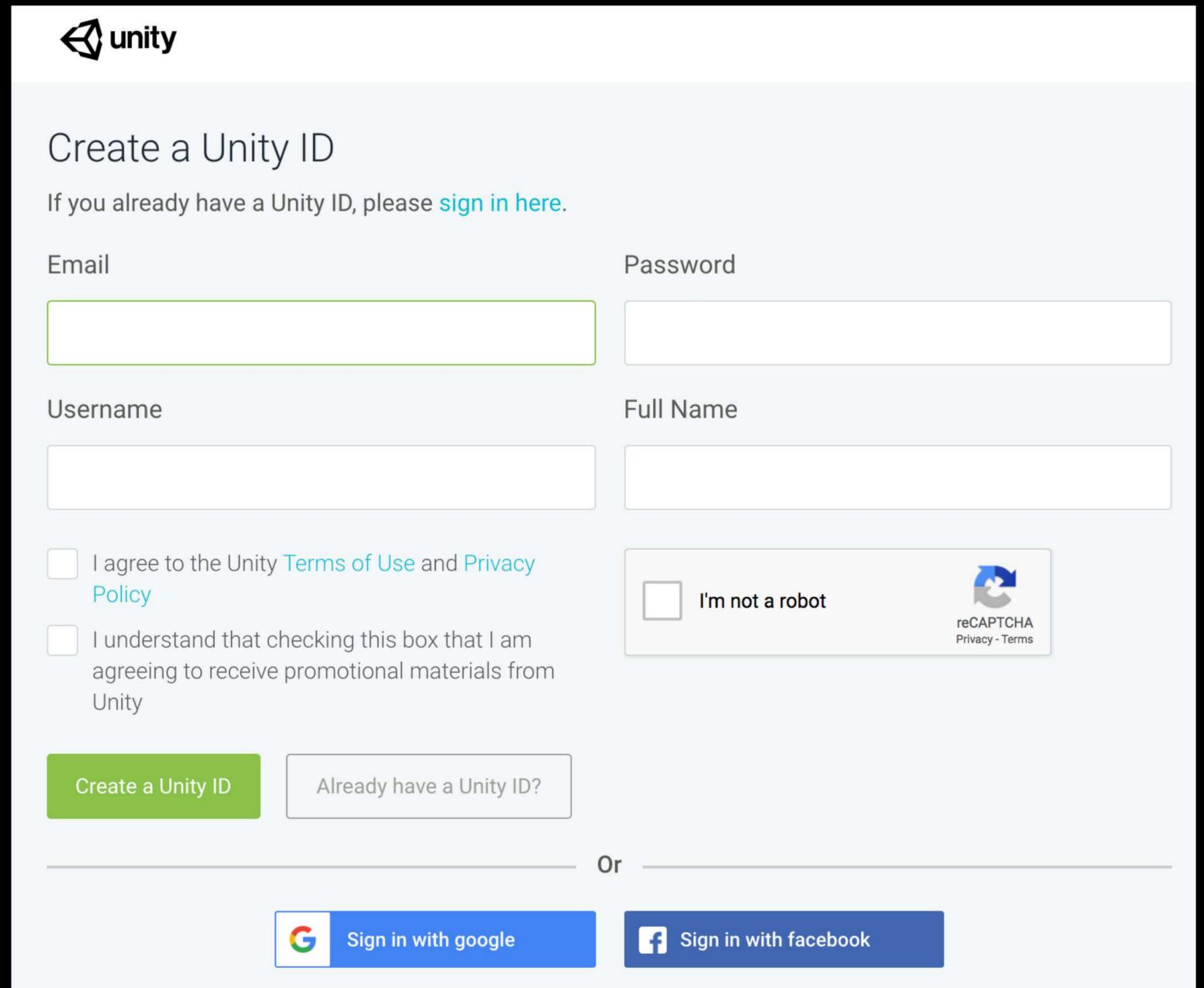
Unity ID是您使用所有Unity服务的身份识别ID。

如果您还没有Unity ID, 请注册:

<https://id.unity.com/>

注册需要您:

- 同意Unity服务条款
- 通过I'm not a robot验证

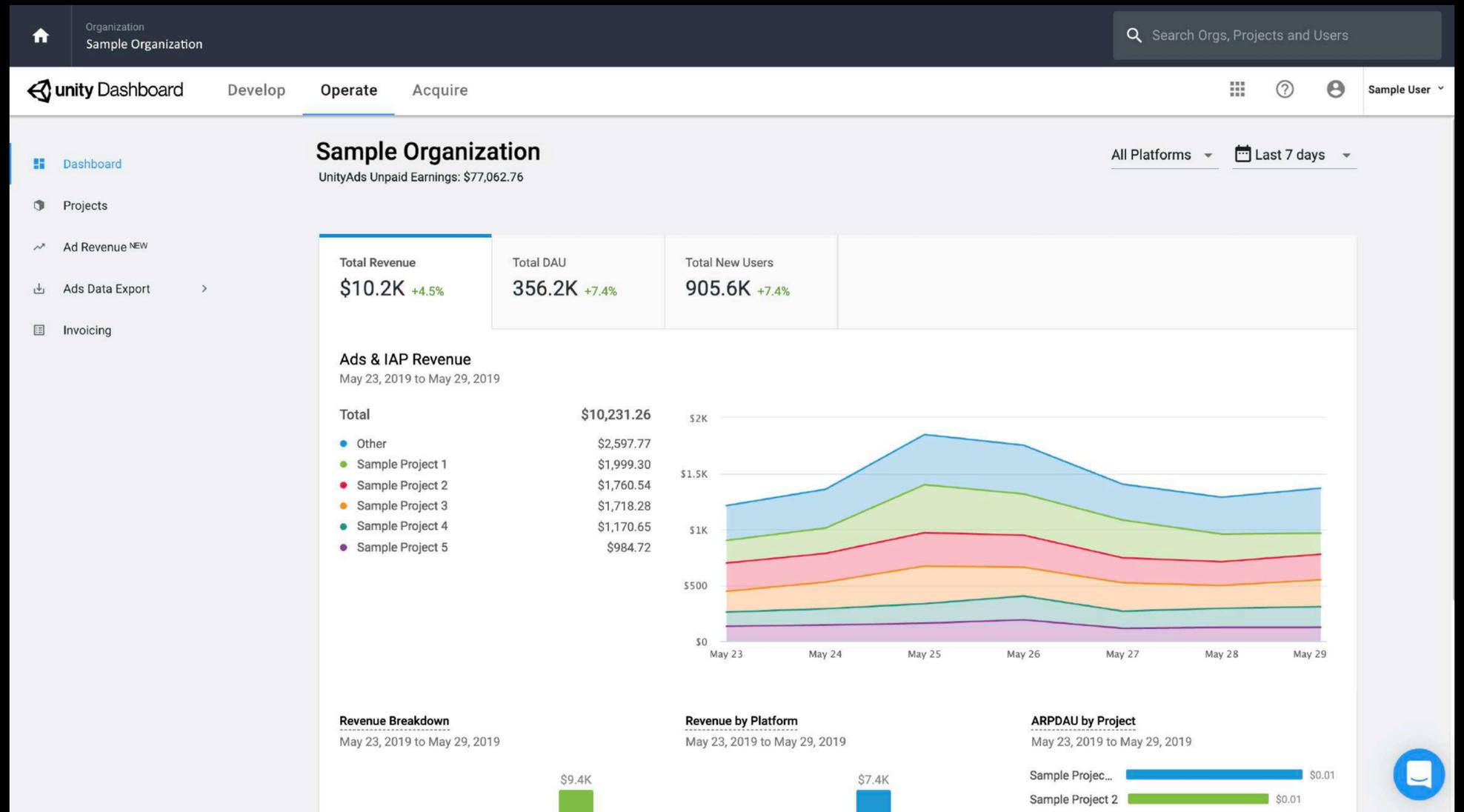


The screenshot shows the Unity ID registration page. At the top left is the Unity logo. The main heading is "Create a Unity ID". Below it, a link says "If you already have a Unity ID, please [sign in here](#)." The form consists of several input fields: "Email" (with a green border), "Password", "Username", and "Full Name". Below these are two checkboxes: "I agree to the Unity [Terms of Use](#) and [Privacy Policy](#)" and "I understand that checking this box that I am agreeing to receive promotional materials from Unity". To the right of these is a reCAPTCHA widget with the text "I'm not a robot" and the reCAPTCHA logo. At the bottom, there are two buttons: "Create a Unity ID" (green) and "Already have a Unity ID?" (white with a grey border). Below these is a horizontal line with "Or" in the center. At the very bottom, there are two social login buttons: "Sign in with google" (with the Google logo) and "Sign in with facebook" (with the Facebook logo).

Operate后台

您可以在Operate后台进行广告位管理，同时我们也为您提供丰富的收入分析报表。

<https://operate.dashboard.unity3d.com/>



Game ID

在您创建Project时，Unity会自动为您生成Game ID.

Game ID是连接SDK与后台的桥梁,广告运营人员需要把这个Game ID提供给开发人员,用于UnityAds SDK的初始化.

Game ID也是Unity唯一定位您游戏的身份识别ID, 向Unity支持咨询问题时, 也需要您提供GameID, 以便针对游戏为您解决问题.

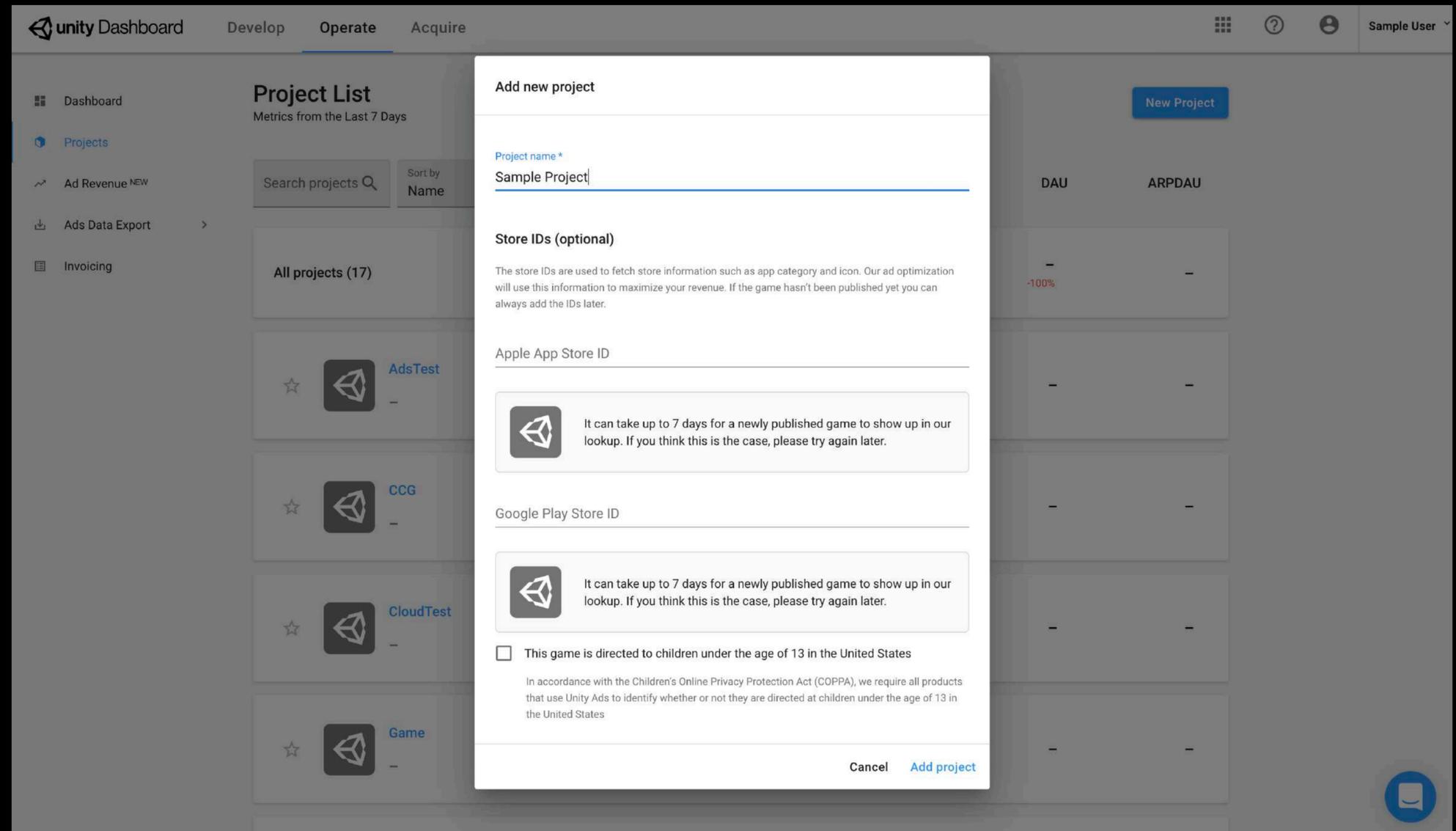
注意:

Unity Game ID由7位数字组成, 它不等同于您游戏的商店应用ID, 请勿混淆使用.

创建Project

- 首先为您的游戏新建一个项目 **NEW PROJECT**.
- 参考说明根据游戏实际情况填写 **Store ID** 和 **COPPA**

提示：
如果您的游戏不是仅供美国13岁以下儿童使用的，则无需勾选COPPA，但我们仍建议您仔细阅读COPPA协议以取得全面的了解。

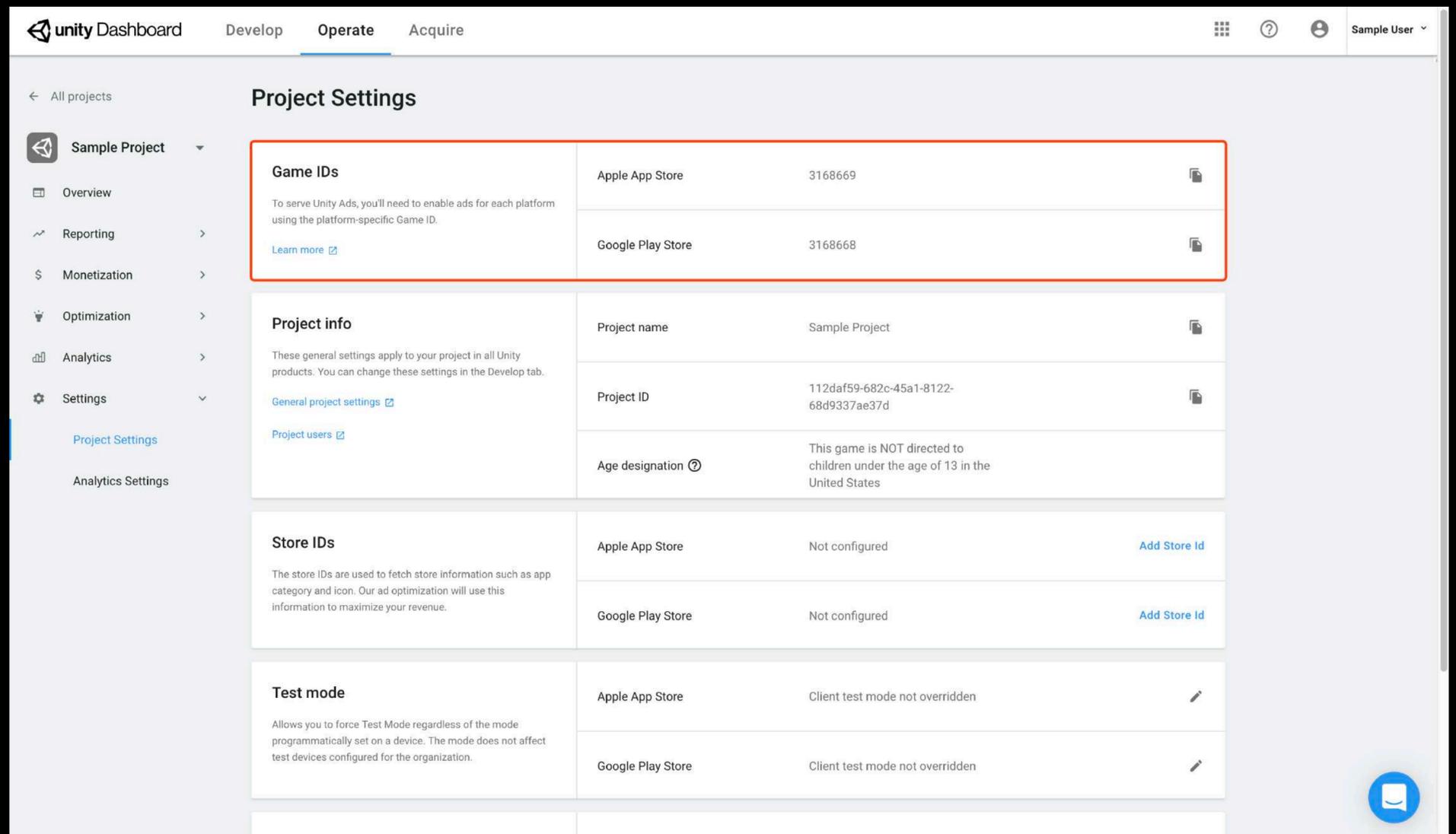


获取Game ID

在后台左侧导航栏

- 选中**Project**.
- 选中**Settings**
- 选中**Project Settings**

即可见Unity自动为该游戏生成的Game ID (7位数字格式).



The screenshot shows the Unity Dashboard interface. The top navigation bar includes 'unity Dashboard', 'Develop', 'Operate', and 'Acquire'. The left sidebar shows a navigation menu with 'Sample Project' selected, and sub-items for 'Overview', 'Reporting', 'Monetization', 'Optimization', 'Analytics', 'Settings', and 'Analytics Settings'. The main content area is titled 'Project Settings' and contains several sections: 'Game IDs', 'Project info', 'Store IDs', and 'Test mode'. The 'Game IDs' section is highlighted with a red border and contains a table with the following data:

Platform	Game ID	Action
Apple App Store	3168669	Copy
Google Play Store	3168668	Copy

Below the 'Game IDs' section, the 'Project info' section shows the Project name as 'Sample Project' and the Project ID as '112daf59-682c-45a1-8122-68d9337ae37d'. The 'Store IDs' section shows that both Apple App Store and Google Play Store are 'Not configured'. The 'Test mode' section shows that the Client test mode is 'not overridden' for both platforms.

Placement ID

在您创建Project时，Unity也会自动为您生成2个广告位Placement ID, 您也可以手动添加更多。

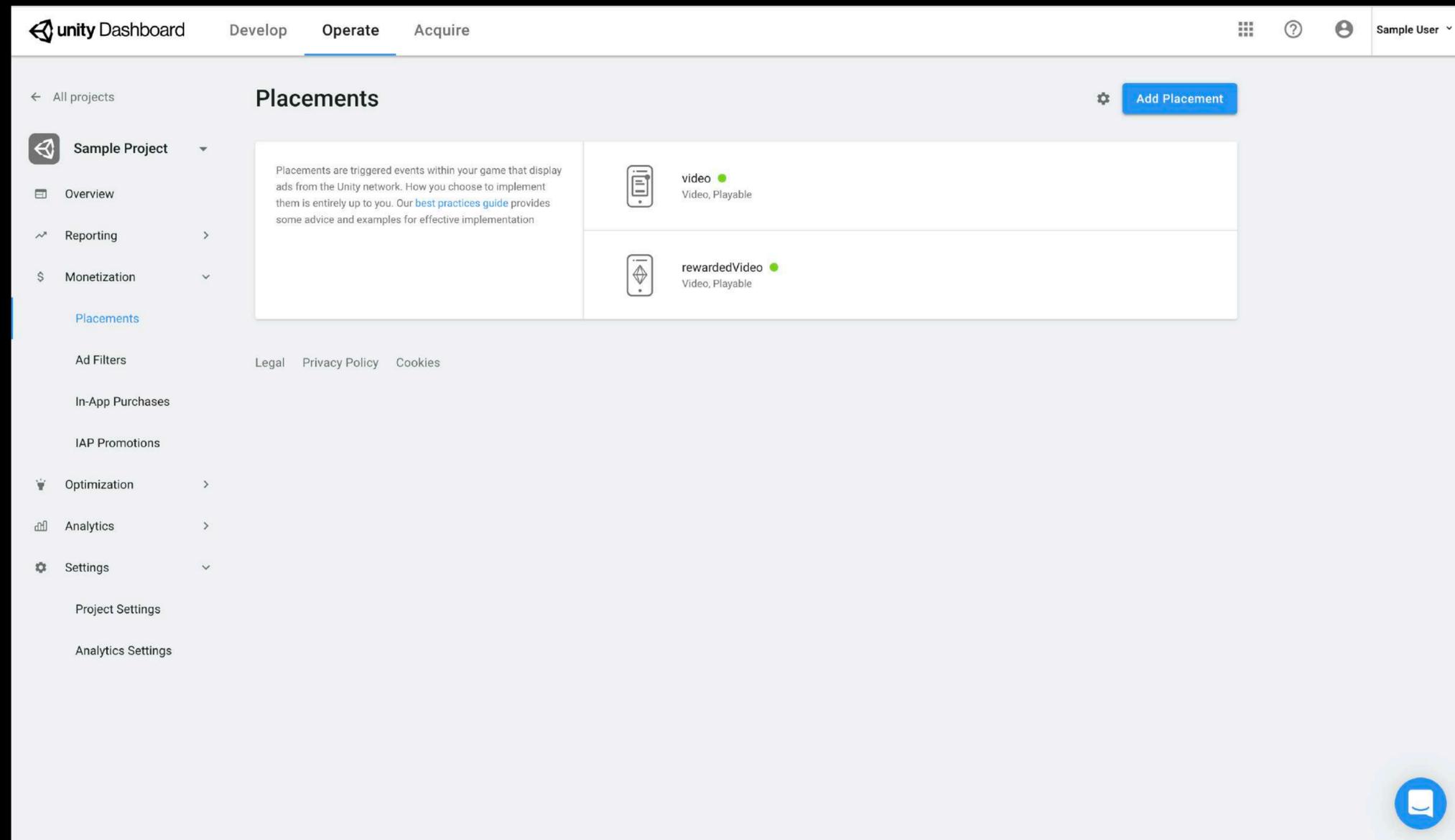
一般开发者可能需要使用不同设置的Placement ID来达到不同的广告效果，这种情况就需要广告运营人员将Placement ID告诉开发人员。

获取Placement ID

在后台左侧导航栏

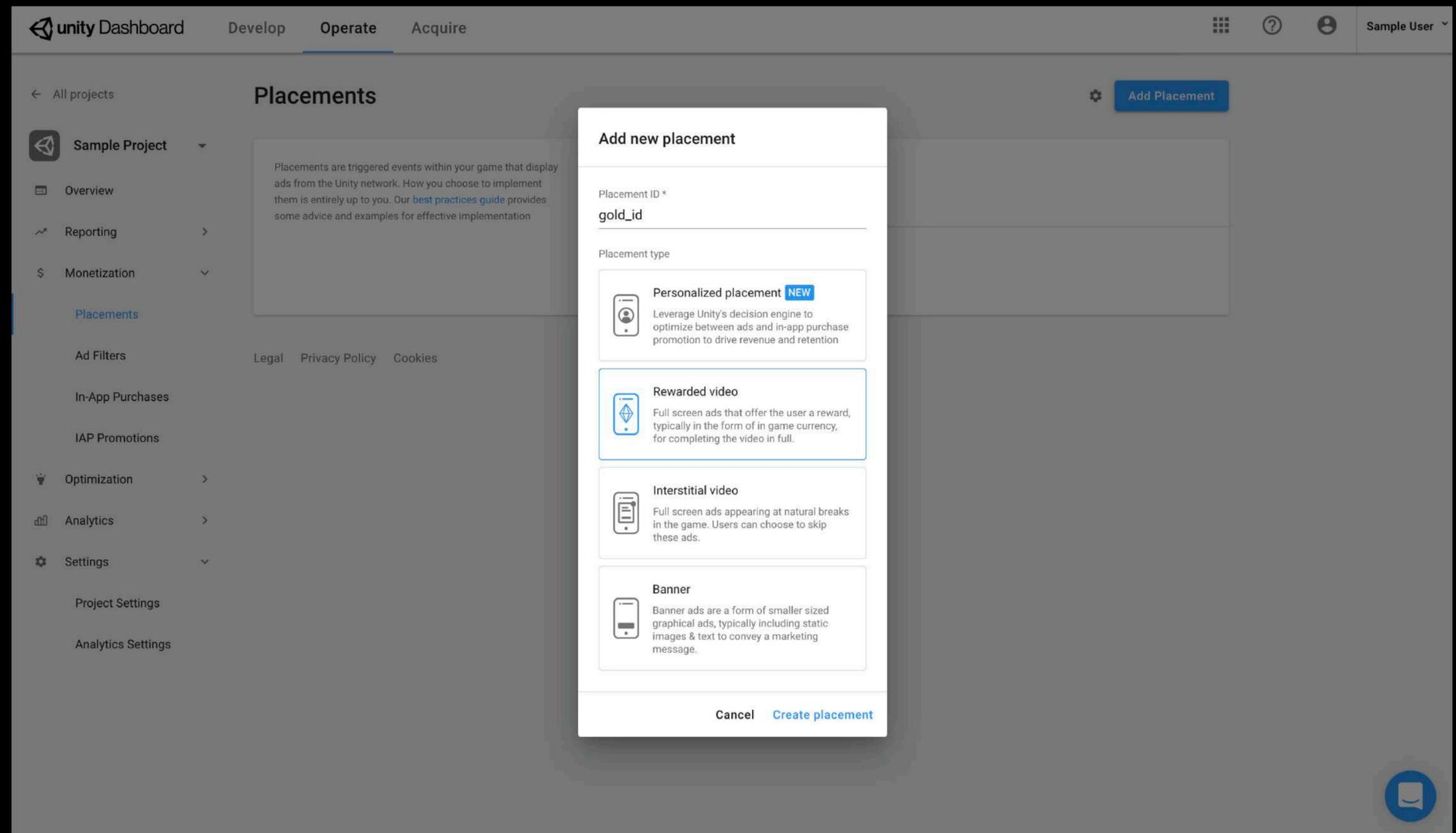
- 选中**Project**.
- 选中**Monetization**
- 选中**Placements**

即可见当前游戏的广告位
Placement ID.



添加Placement

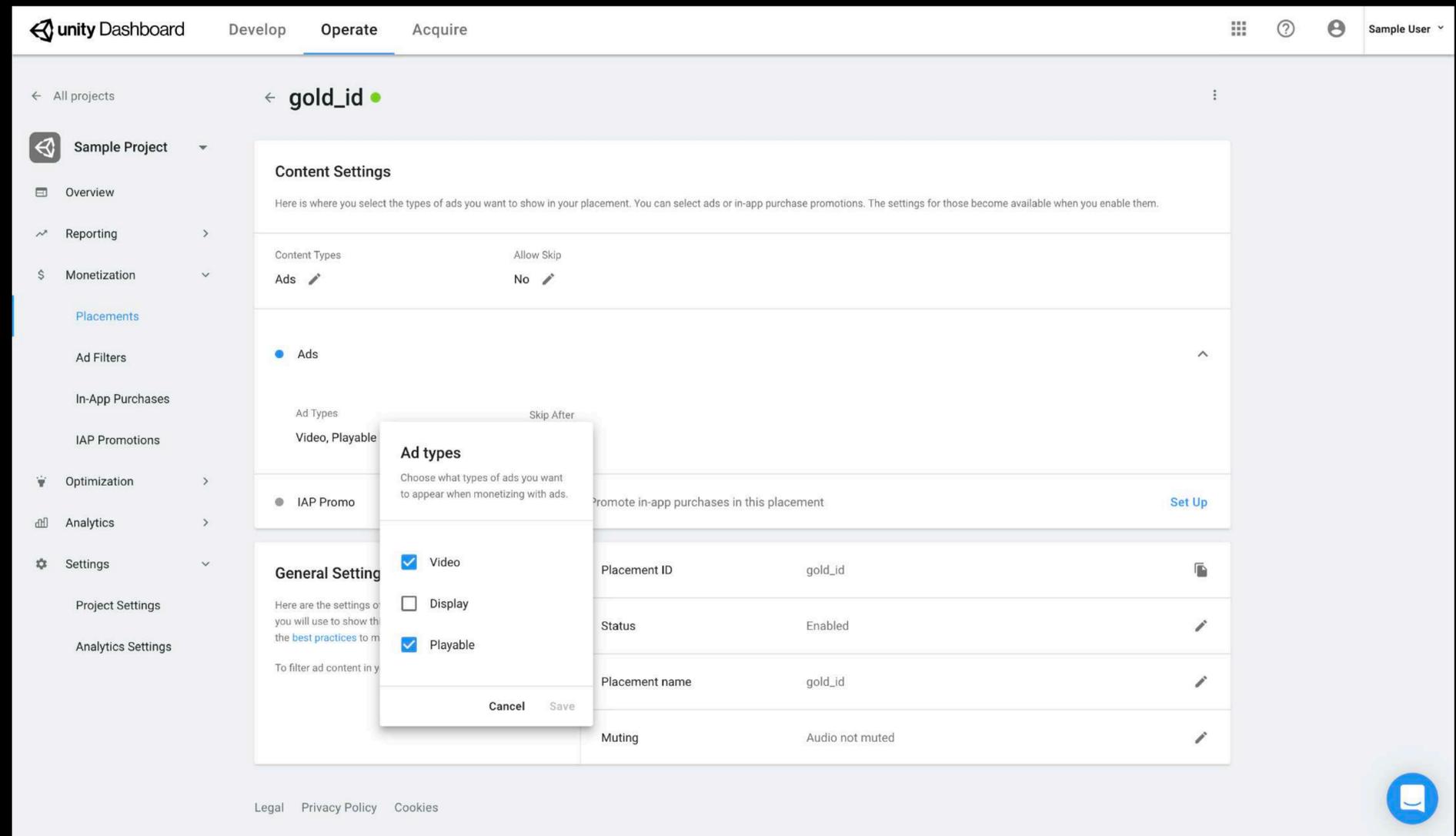
Unity默认会自动为每个游戏生成2个Placement, 您可以通过**ADD PLACEMENT**添加更多.



广告位设置

您可以选中广告位Placement后，在设置页面修改其广告行为，如右图。

提示：
无论是否允许跳过广告，开发人员都能够在视频完整观看完成后给予玩家奖励。
不过根据行业惯例，一般建议激励视频设置为不可跳过。

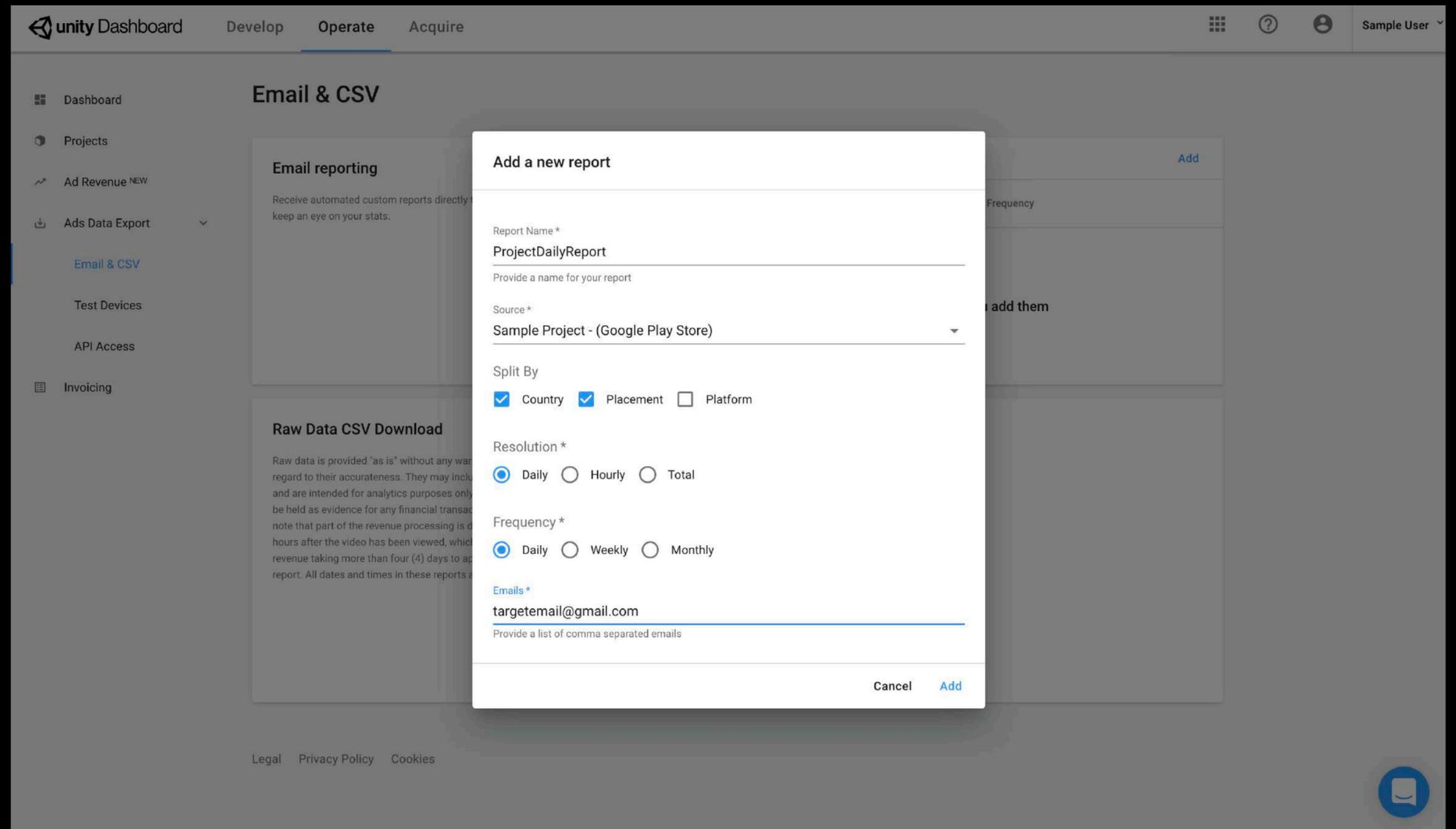


自动发送收入报表

- 选中**Operate**.
- 选中**Ad Data Export**
- 选中**Email & CSV**
- 选中**Add**

提示:

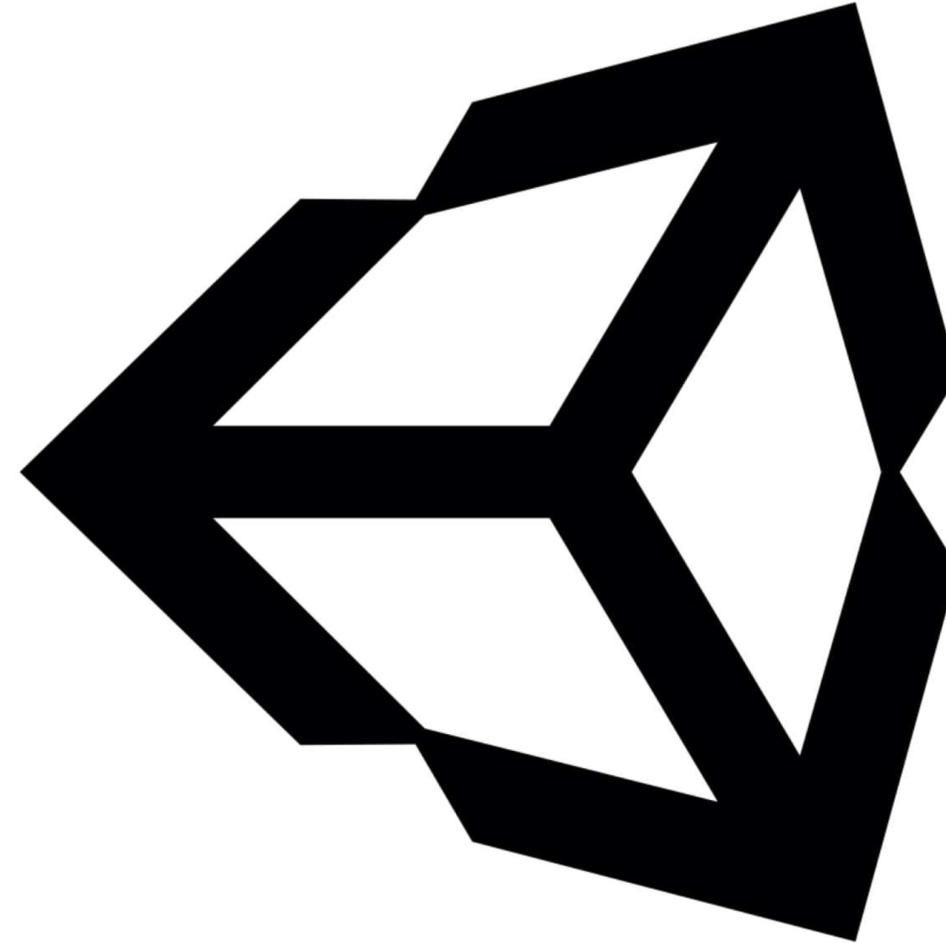
- 支持添加多个报告收件人.
- 如果分割的维度**Split by**较多（尤其同时包含国家与其他维度时），报告可能会很大，或者服务器可能因分析超时(如果服务器处理请求超过60秒，则请求超时)而导致报告无法正常发送.



广告代码集成

Unity支持3种集成方式:

- Asset Store Package
- iOS SDK
- Android SDK



Asset Store Package

使用Unity开发游戏且Unity版本不低于5.6.6的开发者，可以在Unity AssetStore官方账号下载最新的UnityAds资源包：

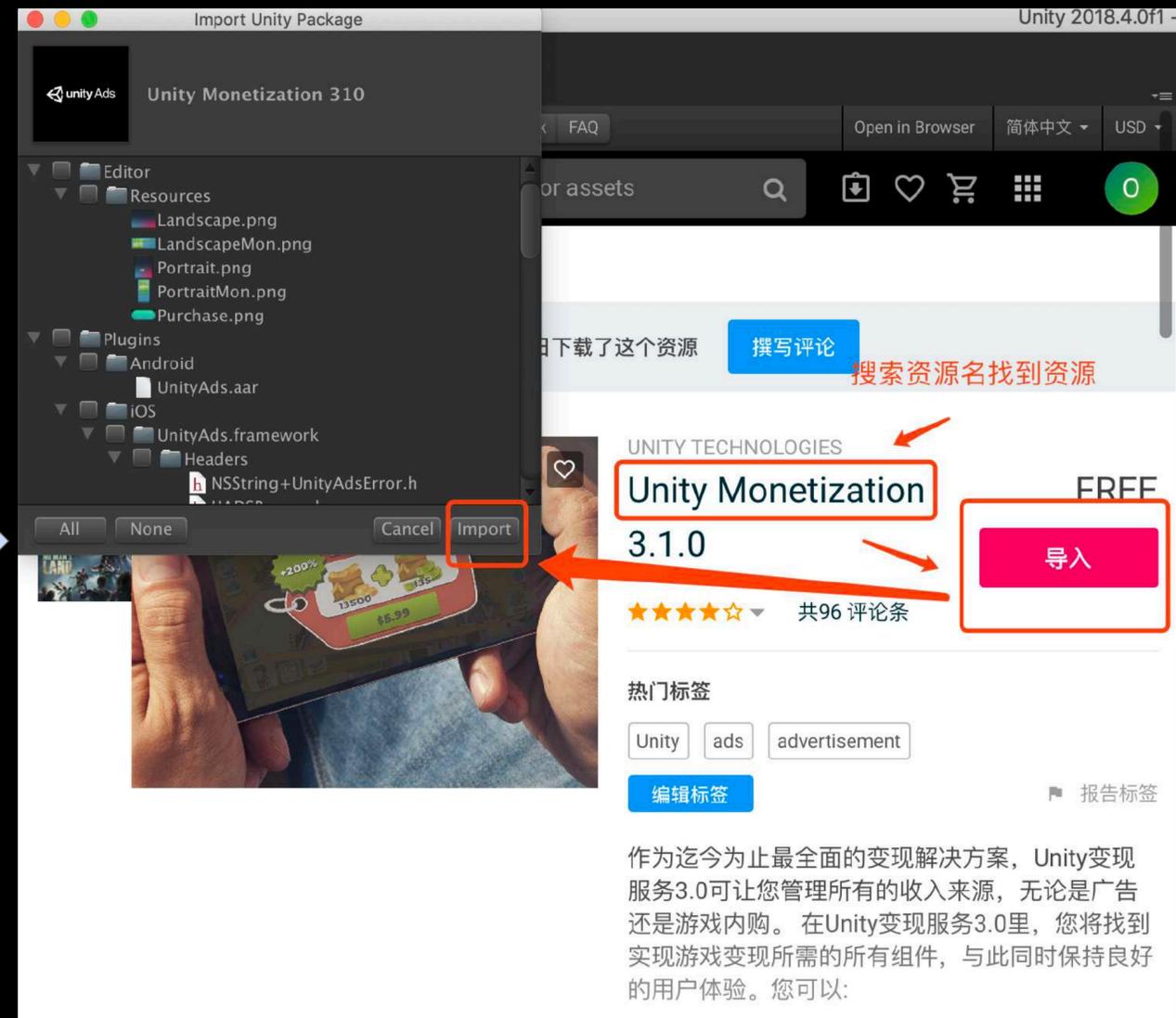
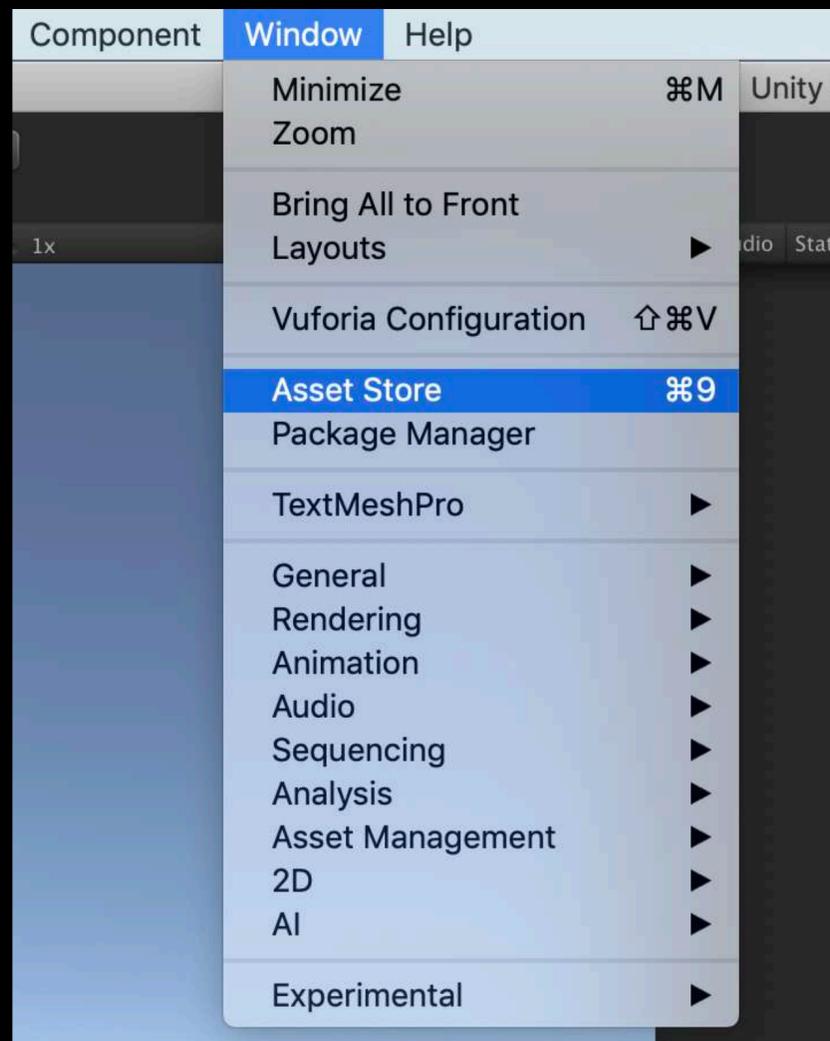
<https://assetstore.unity.com/packages/add-ons/services/unity-ads-66123>

导入广告资源包

您可以从Unity Assets Store导入广告资源包。

提示:

导入后需要到Unity Editor>File>Build Settings, 将Platform切换至iOS或Android.(目前广告服务只支持这2个平台)

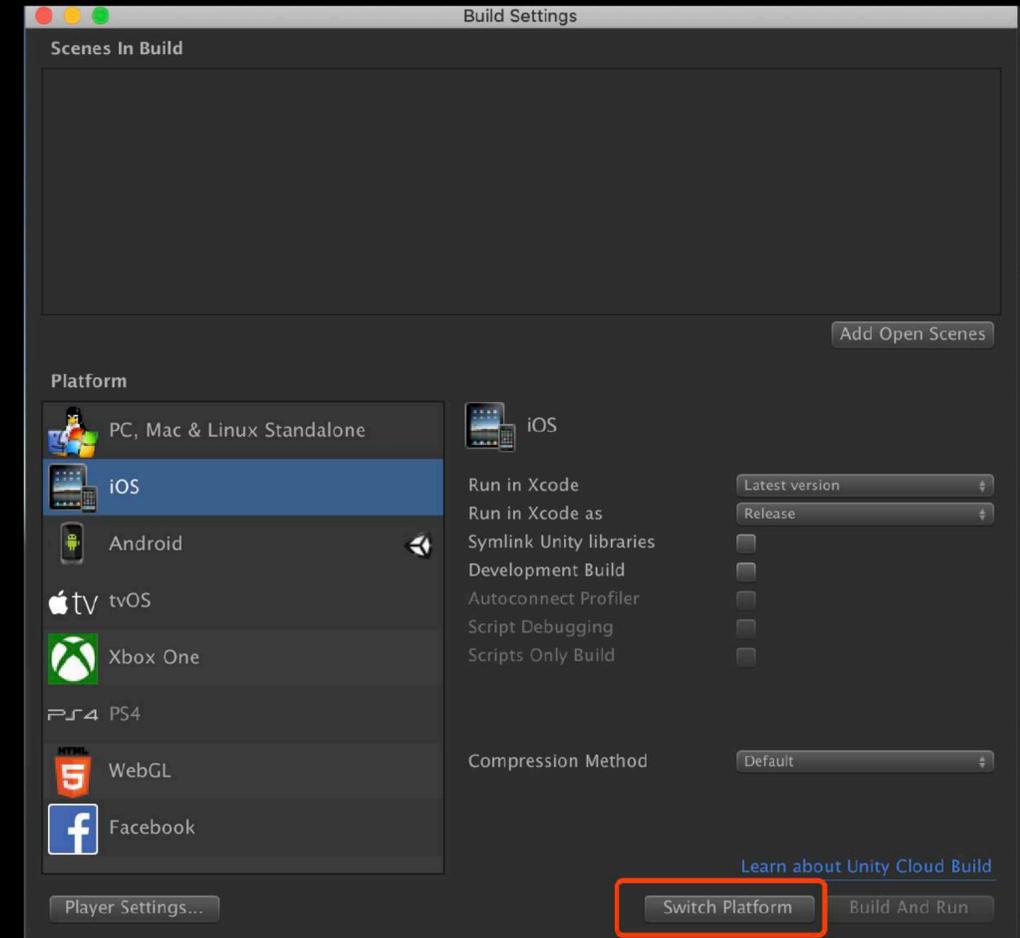
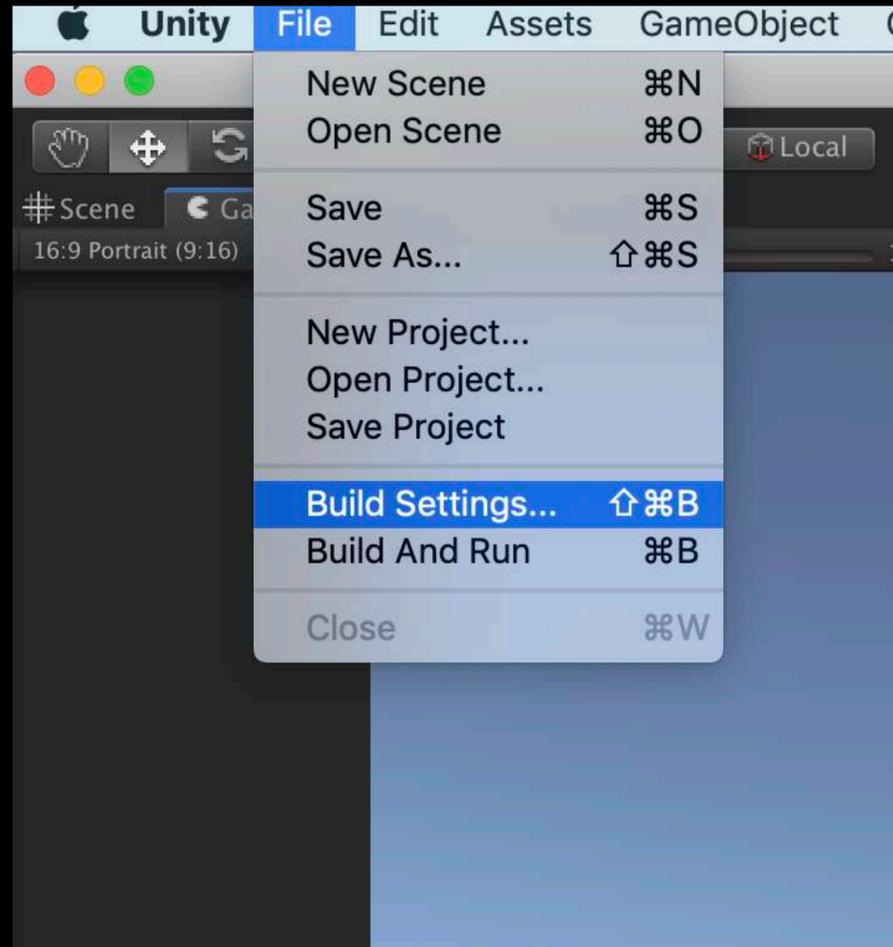


切换平台

到Unity Editor>File>Build Settings, 将Platform切换至iOS或Android.(目前广告服务只支持这2个平台)

*注意通过此方法导入广告资源包后, 不要在Unity编辑器的Services面板上开启Ads服务, Unity默认不开启, 如果您已开启, 请按以下步骤关闭:

点击Editor面板上的  Unity服务图标 -> 点击Ads选项 -> 关闭Ads.



广告展示、回调

- 1.在展示广告 **Show** 之前，需要先用 **IsReady** 判断广告是否已加载完成.
- 2.为了给予玩家奖励，要对广告展示的结果添加监听（需实现 **IUnityAdsListener** 接口）.
- 3.在处理回调时，一般判断在当玩家观看广告完成时给予奖励。

提示:

即使可跳过的广告，如果玩家看完广告的时候，也会触发观看完成的回调。

```
AdsManager.cs
using UnityEngine;
using UnityEngine.Advertisements;

public class AdsManager : MonoBehaviour, IUnityAdsListener
{
    private string gameId = "1234567"; //应用对应平台的Game ID (Dashboard->Operate->选择对应Project->Settings->Project Settings)
    private string placementId = "rewardedVideo"; //Placement ID (Dashboard->Operate->选择对应Project->Monetization->Placements)
    private bool testMode = true; //正式发布前请将测试模式设置为true, 以避免被系统判定涉嫌广告欺诈。

    void Start()
    {
        Advertisement.Initialize(gameId, testMode); //初始化广告SDK, 尽量在应用生命周期的早期进行
        Advertisement.AddListener(this); //添加广告展示结果的监听, 传入的参数this需是一个实现了IUnityAdsListener接口的类
    }

    public void ShowRewardedVideo()
    {
        if (Advertisement.isInitialized && Advertisement.IsReady(placementId))
        {
            Advertisement.Show(placementId);
        }
    }

    /**以下监听***/
    public void OnUnityAdsDidError(string message)
    {
        Debug.Log("OnUnityAdsDidError" + message);
    }

    public void OnUnityAdsDidFinish(string placementId, ShowResult showResult)
    {
        if (showResult == ShowResult.Finished)
        {
            Debug.Log("玩家已经观看完广告, 可在此奖励玩家");
        }
        else if (showResult == ShowResult.Skipped)
        {
            Debug.Log("玩家选择跳过了广告");
        }
        else if (showResult == ShowResult.Failed)
        {
            Debug.Log("广告展示失败");
        }
    }

    public void OnUnityAdsDidStart(string placementId)
    {
        Debug.Log("OnUnityAdsDidStart");
    }

    public void OnUnityAdsReady(string placementId)
    {
        Debug.Log("OnUnityAdsReady, placementId:" + placementId);
    }
}
```

ios SDK

在iOS(Xcode)项目环境下的开发者，可以在Unity官方GitHub账号下载最新的iOS SDK (UnityAds.framework)，使用Objective-C或Swift进行广告集成：

<https://github.com/Unity-Technologies/unity-ads-ios/releases>

SDK 最低支持 iOS 7.0

此处仅讲解Obj-C方法，Swift代码请参考[在线文档](#)。



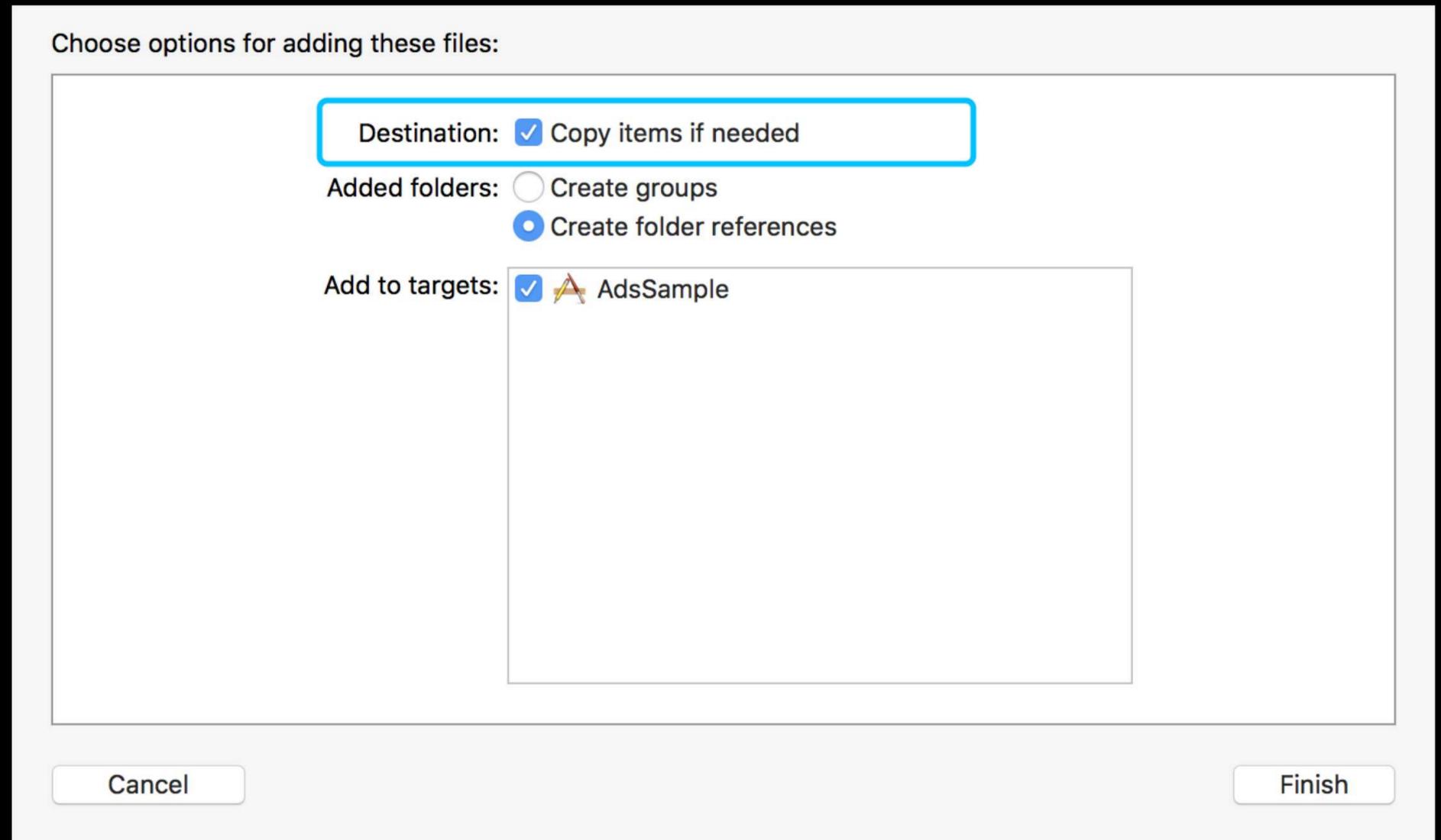
导入广告SDK

首先拖拽UnityAds.framework到游戏项目，选择Copy.

然后在ViewController.h里import UnityAds库.

提示:

如果没用到delegate方法可以不用conform <UnityAdsDelegate>



```
#import <UIKit/UIKit.h>
#import "UnityAds/UnityAds.h"

@interface GameViewController : UIViewController<UnityAdsDelegate>
@end
```

广告初始化

使用UnityAds.initialize() 初始化SDK。将您从开发者后台获取的 Game ID 以String类型，以及代表ViewController的self 作为参数传入。

```
#import "GameViewController.h"
#import "GameScene.h"

@implementation GameViewController
//您需要实现处理广告回调的UnityAdsDelegate接口, 并将其作为initialize方法中的参数引用
//在游戏的运行时生命周期的早期初始化SDK
- (void) viewDidLoad {
    [super viewDidLoad];
    [UnityAds setDebugMode:true]; //调试选项: setDebugMode查看Log
    [UnityAds initialize : @"1234567" delegate : self testMode : true]; // "1234567"是game id, testMode: 是否广告测试模式
}

// Implement the ads listener callback methods:
- (void)unityAdsReady:(NSString *)placementId {
    // Perform logic for ads being available to show.
}

- (void)unityAdsDidStart:(NSString *)placementId {
    // Perform logic for a user starting an ad.
}

- (void)unityAdsDidFinish:(NSString *)placementId
    withFinishState:(UnityAdsFinishState)state {
    // Perform logic for a user finishing an ad.
}

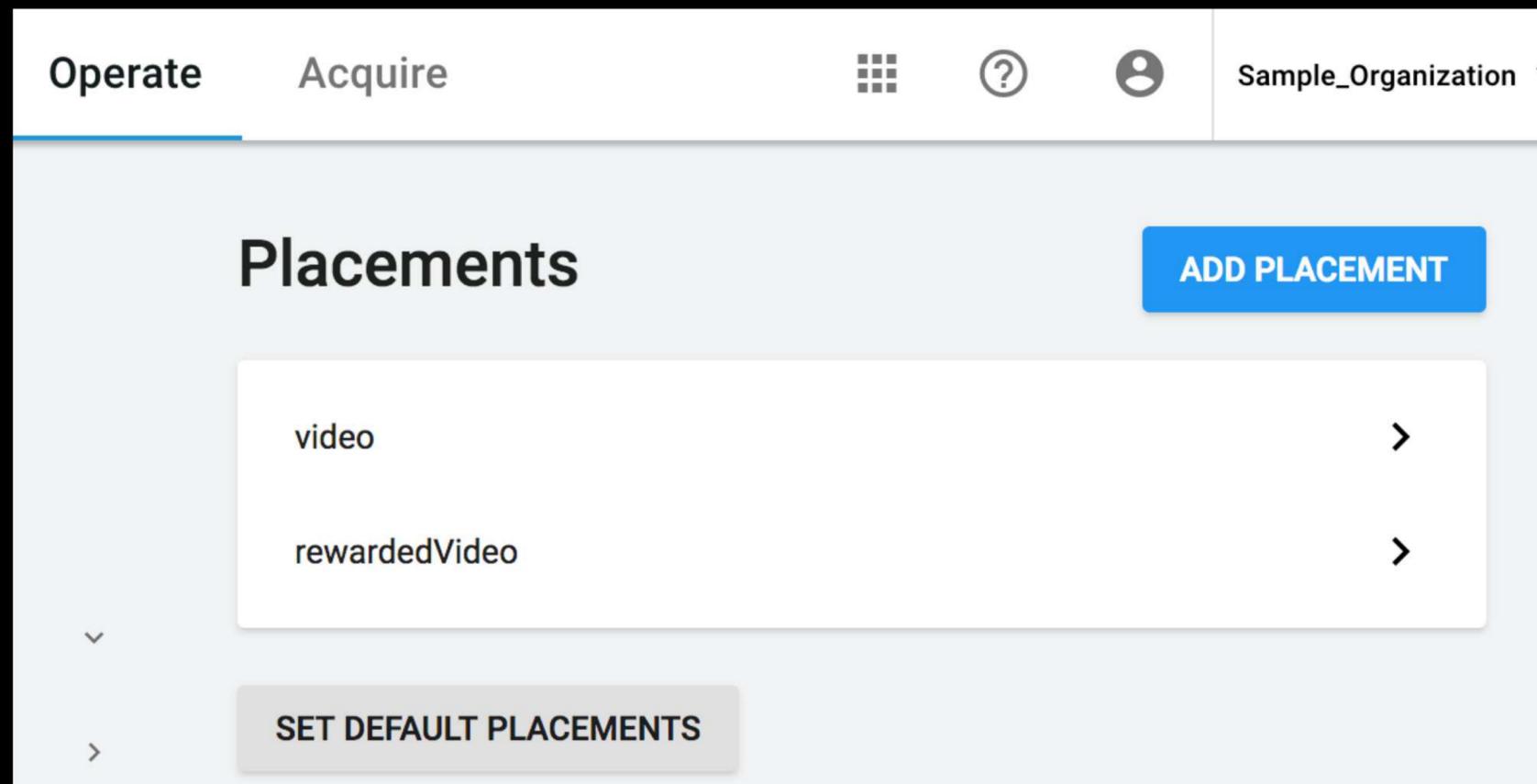
- (void)unityAdsDidError:(UnityAdsError)error withMessage:(NSString *)message {
    // Perform logic for a Unity Ads service error.
}
@end
```

广告展示

使用Placement ID展示广告

提示:

- 展示广告 **show** 之前，需要先用 **isReady** 判断广告是否已加载完成.
- 如果广告始终无法ready，可以使用 **getPlacementState** 来检查广告位的状态.
- 在上线后，仍然可以在dashboard调整广告位是否允许跳过等设置.



```
- (IBAction)showAd:(UIButton *)sender {
    if ([UnityAds isReady:@"rewardedVideo"]) {
        [UnityAds show:self placementId:@"rewardedVideo"];
    }
    else {
        //使用 getPlacementState 检查广告状态
    }
}
```

广告回调

通过实现<UnityAdsDelegate>的方法，可以指定系统的广告回调函数：

提示：

- 一般判断当玩家观看广告完成**completed**时给予奖励。
- 即使可跳过的广告，如果玩家看完广告的时候，也会触发观看完成的回调。

```
- (void)unityAdsDidFinish:(NSString *)placementId
    withFinishState:(UnityAdsFinishState)state {
    // Perform logic for a user finishing an ad.
    switch (state) {
        case kUnityAdsFinishStateError:
            NSLog(@"广告播放出错");
            break;
        case kUnityAdsFinishStateSkipped:
            NSLog(@"用户跳过广告");
            break;
        case kUnityAdsFinishStateCompleted:
            if([placementId isEqualToString:@"rewardedVideo"])
                NSLog(@"广告观看完成，在此奖励用户");
            break;
        default:
            break;
    }
}
```

Android SDK

在Android Studio项目环境下的开发者，可以在Unity官方GitHub账号下载最新的Android SDK (unity-ads.aar)进行广告集成：

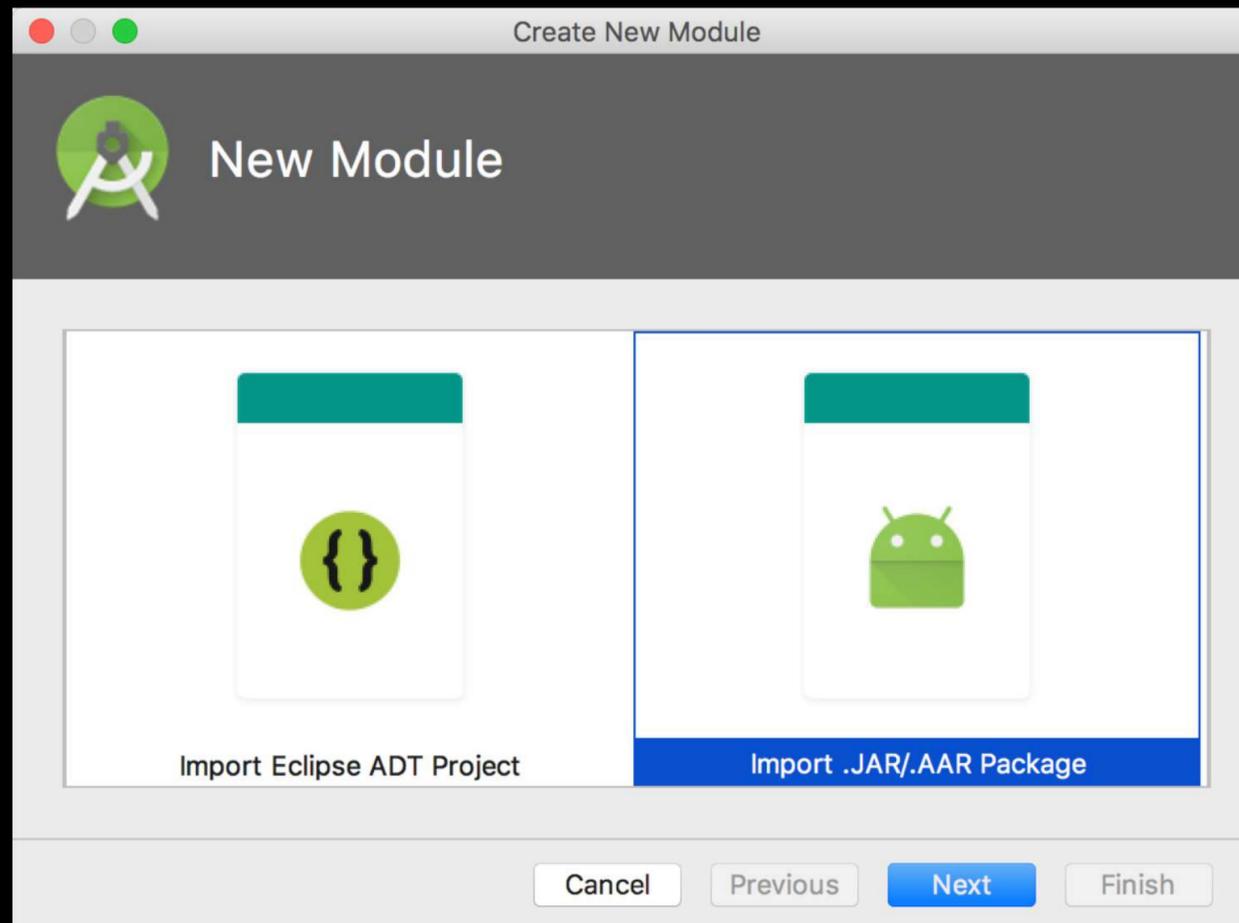
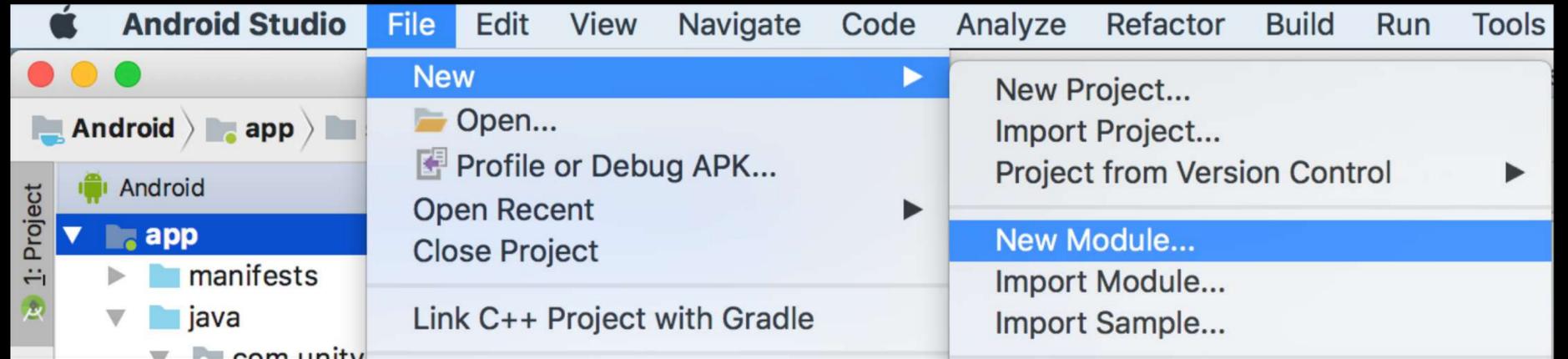
<https://github.com/Unity-Technologies/unity-ads-android/releases>

SDK 最低支持 API 16 (Android 4.1)

无法使用.aar的开发者则需参考在线文档[integrating without Android Studio](#)部分进行广告集成。

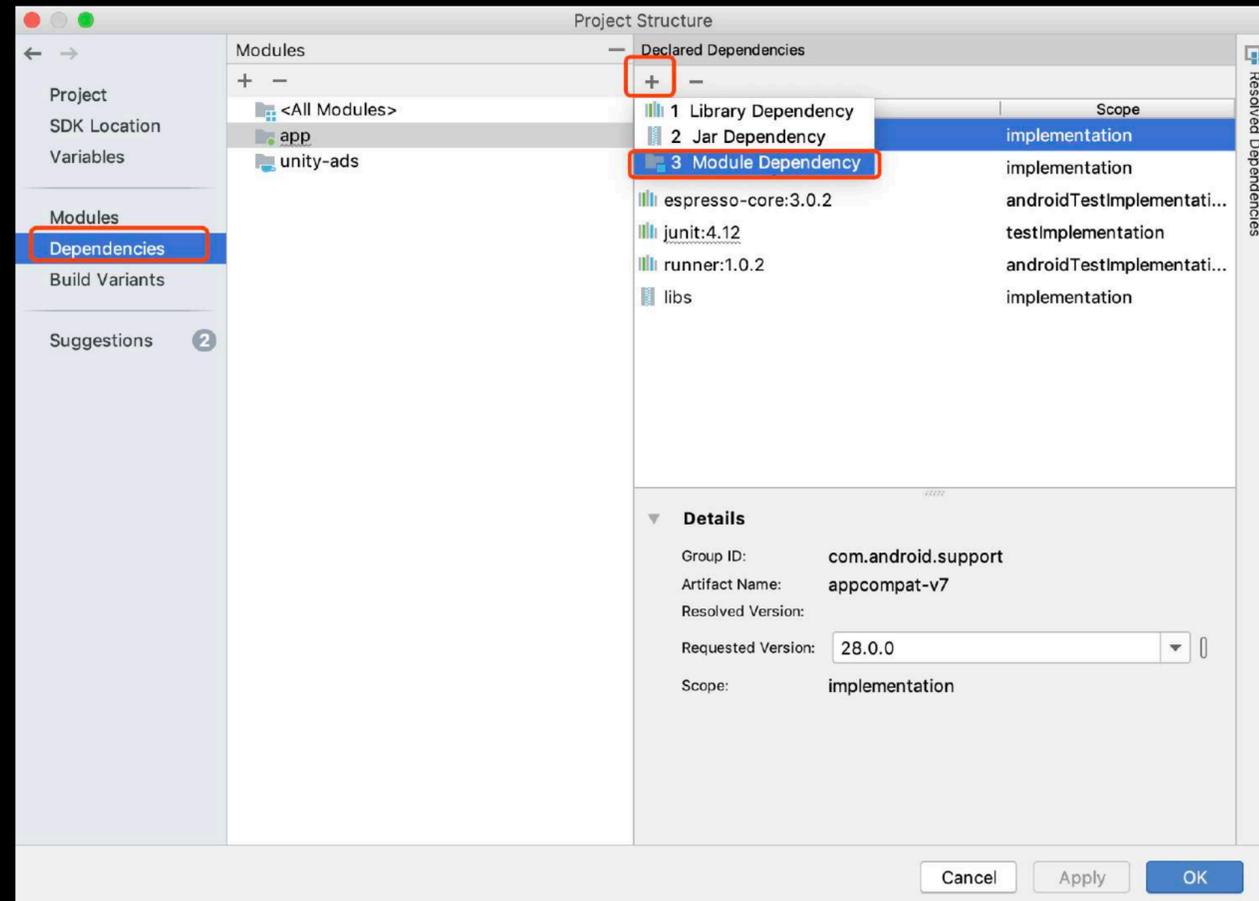
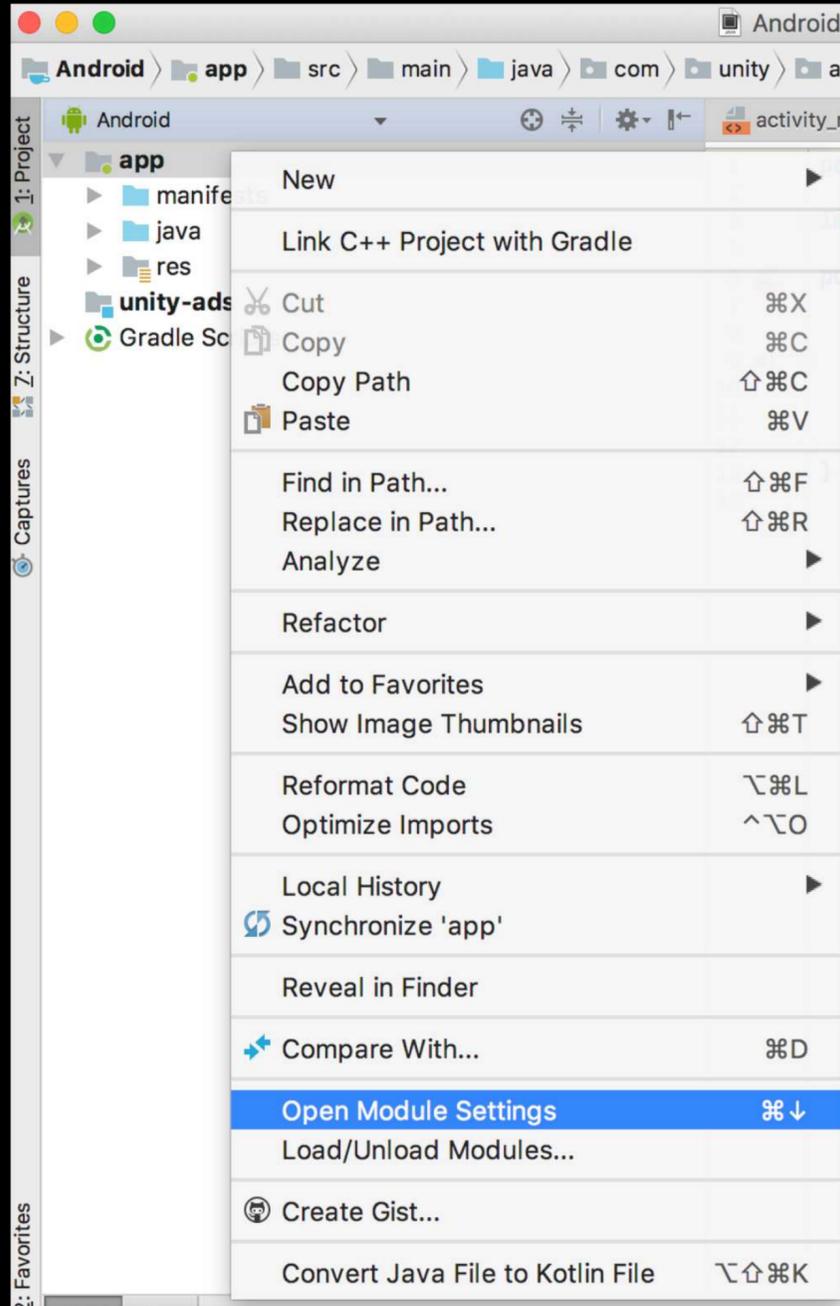
导入广告SDK(1/2)

新建aar类型Module，选择unity-ads.aar导入到项目中。



导入广告SDK(2/2)

在app的Module Settings里，添加unity-ads依赖。



实现接口

在Activity 中导入UnityAds 和 IUnityAdsListener.

implements IUnityAdsListener.

Android Studio 会提示您缺少实现接口所必须添加的回调方法，点击提示后选择Implement methods 一键添加; 当然您也可以手动添加这些回调方法.

```
package com.unity.unityads;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import com.unity3d.ads.IUnityAdsListener;
import com.unity3d.ads.UnityAds;

public class MainActivity extends AppCompatActivity implements IUnityAdsListener {

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

    @Override
    public void onUnityAdsReady(String s) {

    }

    @Override
    public void onUnityAdsStart(String s) {

    }

    @Override
    public void onUnityAdsFinish(String s, UnityAds.FinishState finishState) {

    }

    @Override
    public void onUnityAdsError(UnityAds.UnityAdsError unityAdsError, String s) {

    }
}
```

广告初始化

使用Game ID初始化广告

调试选项:

setDebugMode查看Log

testMode使用测试广告

重要!

请只初始化一次。

Unity SDK 不支持多进程，且所有广告操作要放在主线程。

```
package com.unity.unityads;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import com.unity3d.ads.IUnityAdsListener;
import com.unity3d.ads.UnityAds;

public class MainActivity extends AppCompatActivity implements IUnityAdsListener{

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

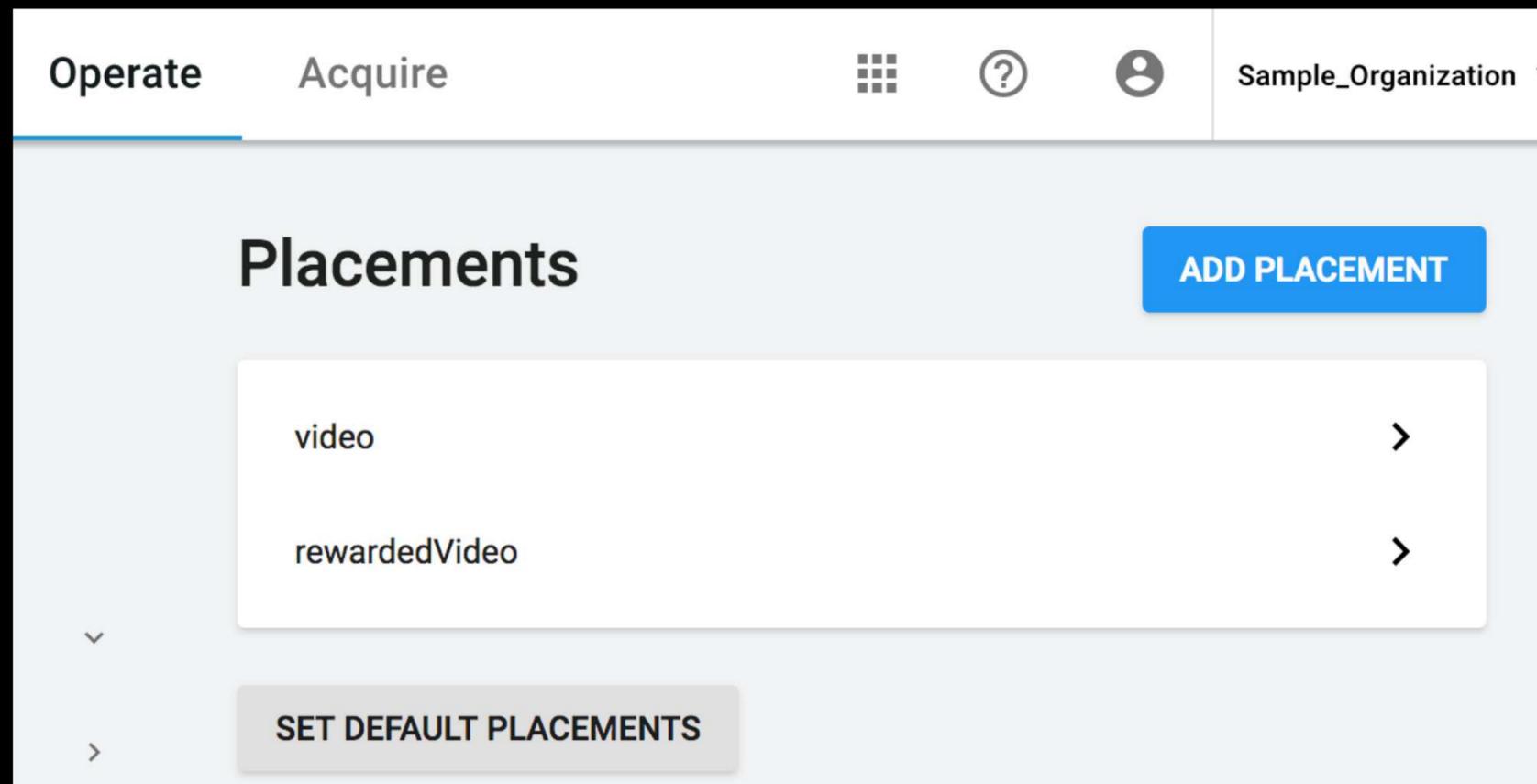
        UnityAds.setDebugMode(true); //查看log
        UnityAds.initialize(activity: this, gameId: "1234567", listener: this, testMode: true);
    }
}
```

广告展示

使用Placement ID展示广告

提示:

- 展示广告 **show** 之前，需要先用 **isReady** 判断广告是否已加载完成.
- 如果广告始终无法ready，可以使用 **getPlacementState** 来检查广告位的状态.
- 在上线后，仍然可以在dashboard调整广告位是否允许跳过等设置.



```
public void showAd(View view) {  
    if (UnityAds.isReady(placementId: "rewardedVideo")) {  
        UnityAds.show(activity: this, placementId: "rewardedVideo");  
    }  
    else {  
        Log.d(tag: "Unity Sample", msg: "UADS is not Ready");  
        //使用 getPlacementState 检查广告状态  
    }  
}
```

广告回调

通过实现IUnityAdsListener的方法，可以指定广告回调函数：

提示：

- 一般判断当玩家观看广告完成**completed**时给予奖励。
- 即使可跳过的广告，如果玩家看完广告的时候，也会触发观看完成的回调。

```
@Override
public void onUnityAdsReady(String placementID) {
    Log.d( tag: "Unity Sample", msg: "UADS Ready");
}

@Override
public void onUnityAdsStart(String placementID) {
    Log.d( tag: "Unity Sample", msg: "UADS Start");
}

@Override
public void onUnityAdsFinish(String placementID, UnityAds.FinishState finishState) {
    switch (finishState) {
        case ERROR:
            Log.d( tag: "Unity Sample", msg: "UADS finished with error");
            break;
        case SKIPPED:
            Log.d( tag: "Unity Sample", msg: "UADS was skipped");
            break;
        case COMPLETED:
            Log.d( tag: "Unity Sample", msg: "UADS was completed");
            if (placementID.equals("rewardedVideo")) {
                Log.d( tag: "Unity Sample", msg: "Offer a reward to the player");
            }
            break;
        default:
            break;
    }
}

@Override
public void onUnityAdsError(UnityAds.UnityAdsError unityAdsError, String message) {
    Log.d( tag: "Unity Sample", msg: "UnityAds ERROR: " + unityAdsError + message);
}
```

观看完毕，奖励用户

附录

- 填写Store ID
- Test Mode
- 获得技术支持
- 资源链接



填写Store ID -iOS

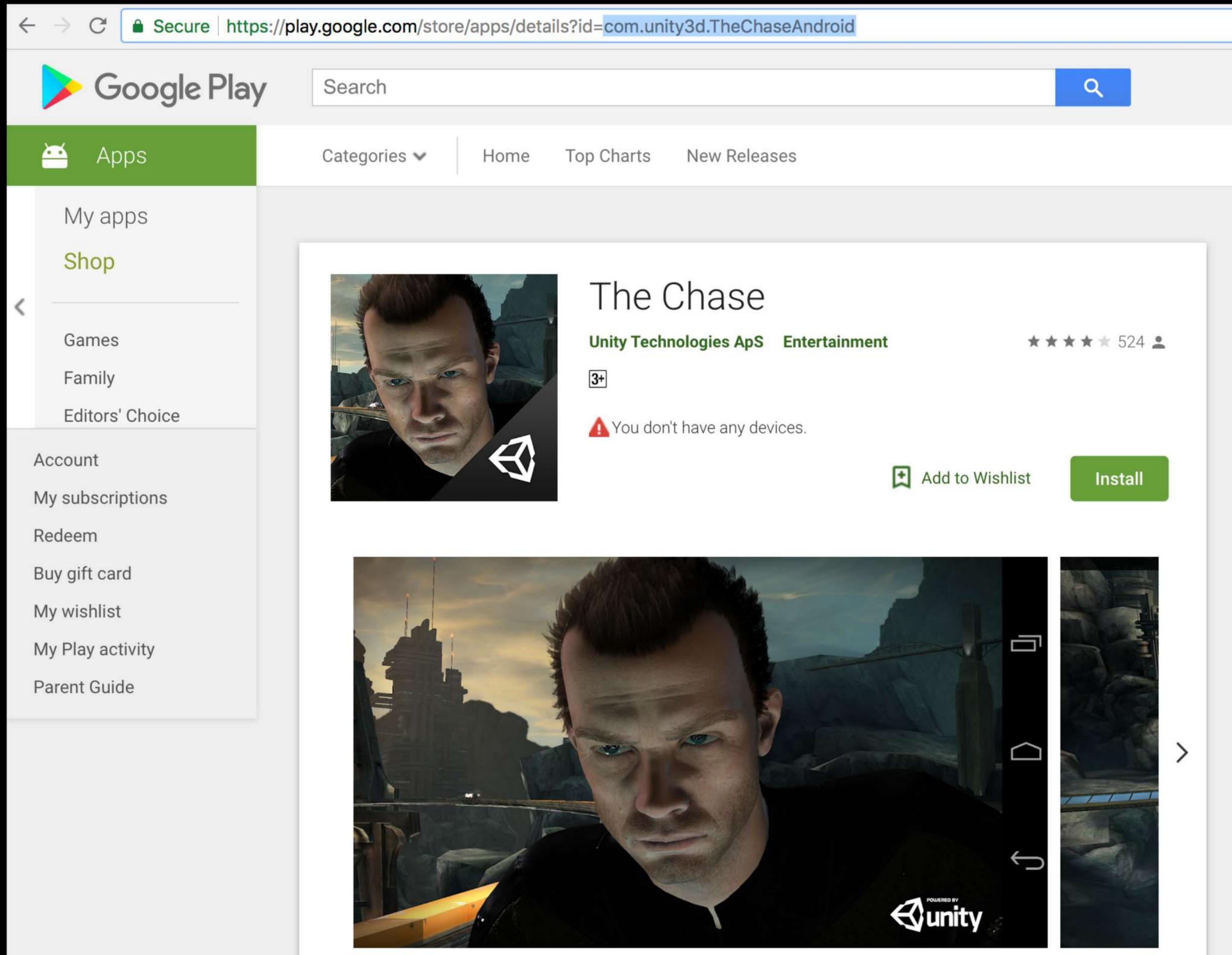
iOS游戏的Store ID
在APP的iTunes
store URL中可以找
到，如右图高亮部分
：

The screenshot shows the iTunes Store page for the app 'Unity Chase'. The URL in the browser's address bar is `https://itunes.apple.com/us/app/unity-chase/id820347596?mt=8`, with the Store ID '820347596' highlighted in blue. The page features the app's icon, a '4+' age rating, the developer 'Unity Technologies ApS', a 3.4-star rating from 16 reviews, and a 'Free' price tag. Below the app information, there are three screenshots of the game, each with a 'POWERED BY unity' logo in the bottom right corner. The navigation bar at the top includes links for Mac, iPad, iPhone, Watch, TV, Music, and Support.



填写Store ID -Android

Android游戏的Store ID在APP的Google Play store URL中可以找到，如右图高亮部分：



Test Mode

Unity支持通过添加测试设备、平台设置、代码方式获取测试广告。
测试结束后请务必关闭测试广告，以免影响您的变现收入。

优先级: **Test Devices > Platform Settings > 代码**

Test Devices

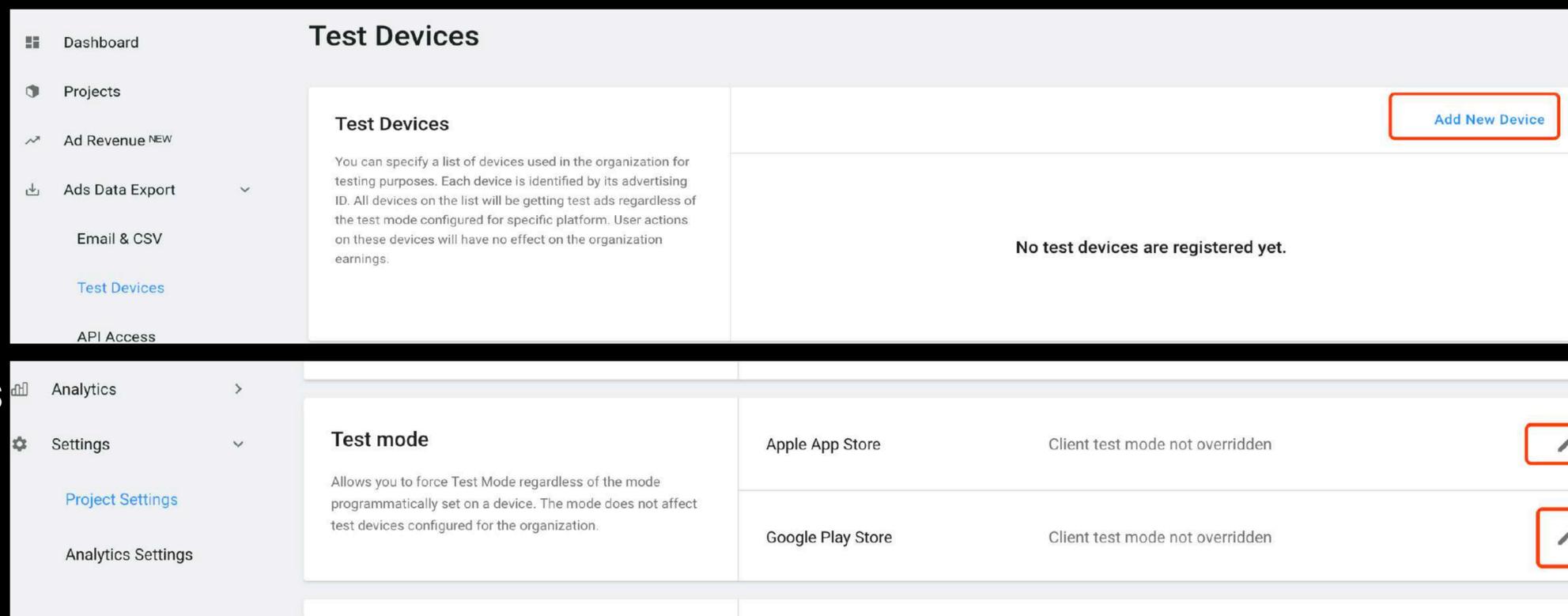
Operate > Ad Data Export > Test Devices

Platform Settings

Operate > Project > Settings > Project Settings

代码

调用初始化方法时可以设置TestMode



获取技术支持

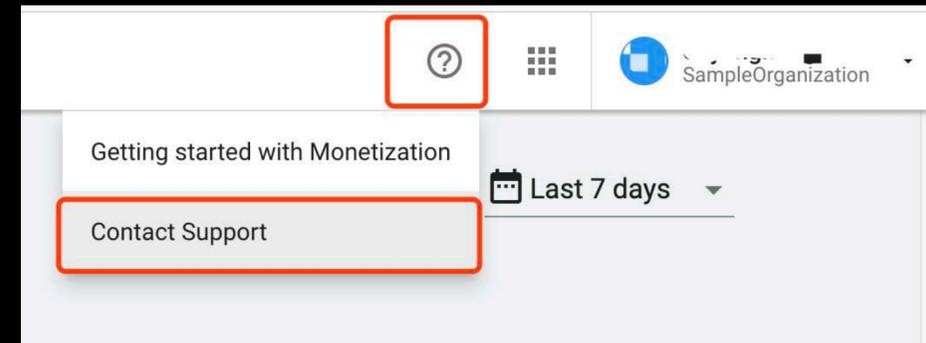
Operate后台:

点击Operate右上角  后选择Contact Support填写表格获得支持，如右图，这种方式我们可以自动查询您的账户信息，以便更快速的为您解决问题。

邮件:

unityads-support@unity3d.com

如果您无法使用Contact Support form，您也可以发邮件获取支持，但需要您提供[详情信息](#)以便问题的定位和解决。



Contact support

Organization *
SampleOrganization

Category *

Subject *

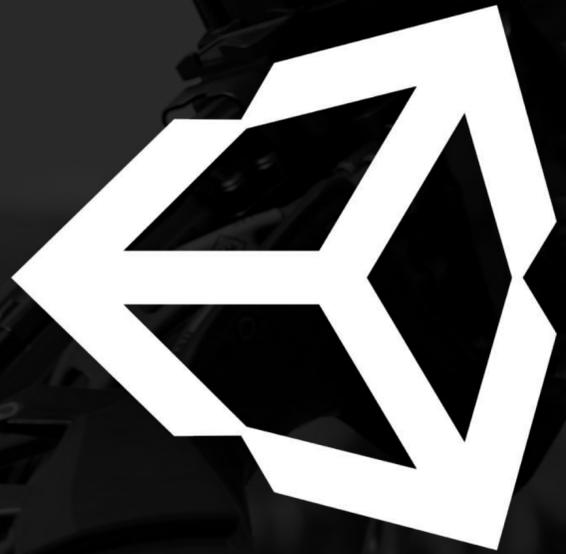
Description *

Attachments
Add screenshots, images, etc. Choose File

Cancel Send

资源链接

- Unity ID: <http://id.unity.com/>
- Operate后台: <https://operate.dashboard.unity3d.com>
- Asset Package 下载: <https://assetstore.unity.com/packages/add-ons/services/unity-ads-66123>
- iOS SDK 下载: <https://github.com/Unity-Technologies/unity-ads-ios/releases>
- Android SDK 下载: <https://github.com/Unity-Technologies/unity-ads-android/releases>
- Unity Ads 在线文档 [英文]: <http://unityads.unity3d.com/help/index>
- Unity Ads 在线文档 [中文]: <https://github.com/unity-cn/unityads-help-cn/wiki>
- 常见问题FAQ: <https://github.com/unity-cn/unityads-help-cn/wiki/monetization>
- 广告技术支持: unityads-support@unity3d.com



Unity Ads Support