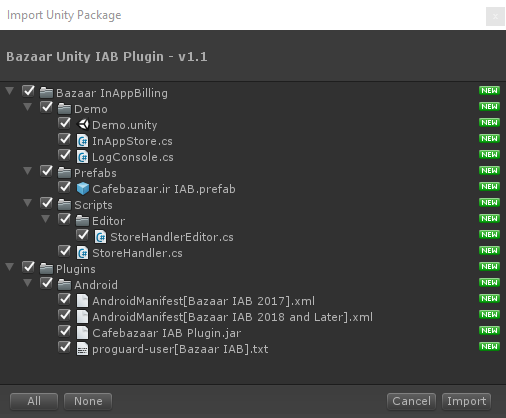
بسمه تعالی

**پلاگین پرداخت کافه‌بازار برای یونیتی (نسخه 1.1)**

**دانلود و اضافه کردن پلاگین**

برای راه‌اندازی و استفاده از پلاگین، این فایل UnityPackage را دانلود کنید. پروژه خود را باز کنید و سپس فایل دانلود شده را اجرا کنید. بعد از بارگزاری شدن پکیج، گزینه import را بزنید و صبر کنید تا فایل‌های پلاگین به پروژه شما اضافه شود.

<https://github.com/hojjat-reyhane/Cafebazaar-IAB-for-Unity/blob/master/Bazaar%20Unity%20IAB%20Plugin%20-%20v1.1.unitypackage>



تصویر ۱: پنجره import کردن پکیج در یونیتی

پکیج بالا قابل اجرا در یونیتی 2017.4.1 و بالاتر است. همچنین می‌توانید فایل‌های پلاگین را به صورت جداگانه از این لینک در github دانلود کنید.

<https://github.com/hojjat-reyhane/Cafebazaar-IAB-for-Unity>

**راه‌اندازی**

در پروژه خود اولین صحنه بازی که اجرا می‌شود را باز کنید. سپس از مسیر Assets/Bazaar InAppBilling/Prefabs در پروژه خود، prefab موجود با نام “Cafebazaar.ir IAB” را بکشید و در صحنه رها کنید. این prefab دارای یک component اصلی به نام “Store Handler” می‌باشد که باید اطلاعات محصولات و پرداخت خود را در آن وارد نمایید.

**نکته:** نیازی به اضافه کردن prefab به تمام صحنه‌ها نیست، تنها اولین صحنه بازی کافی است.

**تنظیمات فایل Manifest**

برای استفاده از سرویس پرداخت بازار نیاز دارید تا permission و اطلاعات Activity پرداخت را در فایل Manifest پروژه خود وارد کنید. نمونه فایل Manifest در مسیر Assets/Plugins/Android برای نسخه‌های 2017 و 2018 به بعد قرار دارد. در صورتی که پروژه شما از قبل فایل Manifest ندارد، می‌توانید مطابق نسخه یونیتی خود از یکی از فایل‌های نمونه استفاده کنید.

**نکته:** دقت داشته باشید که نام فایل Manifest باید کاملا به صورت AndroidManifest.xml باشد و در مسیر Assets/Plugins/Android قرار بگیرد.

همچنین در صورتی که پروژه شما از قبل دارای فایل Manifest می‌باشد، می‌توانید به صورت دستی permission زیر را در فایل Manifest خود اضافه کنید:

<uses-permission android:name="com.farsitel.bazaar.permission.PAY\_THROUGH\_BAZAAR" />

و باید اطلاعات Activity پرداخت را در فایل Manifest درون تگ application و قبل از بسته شدن آن وارد کنید.

<activity

android:name="ir.cafebazaar.iab.ServiceBillingBazaar$IabActivity"

android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />

**تنظیمات Gradle و Proguard**

یونیتی از نسخه ۲۰۱۸ به بعد از سیستم خروجی گرفتن Gradle استفاده میکند. پلاگین پرداخت بازار نیاز به تنظیمات خاصی در Gradle ندارد. اما در صورتی که تنظیمات Gradle پروژه شما از Proguard استفاده می‌کند باید دستور زیر را در فایل Proguard خود وارد کنید:

-keep class ir.cafebazaar.iab.\*\* { \*; }

در صورت اضافه نکردن این دستور، بعد از اجرای بازی بر روی دستگاه و هنگام استفاده از پرداخت با خطا مواجه خواهید شد.

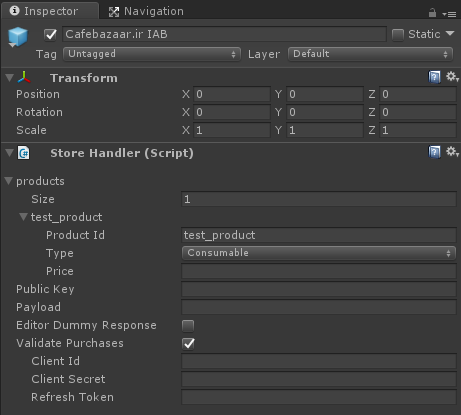
برای ساختن فایل Proguard در یونیتی به بخش Player Settings رفته و در قسمت Publishing Settings گزینه User Proguard File را تیک بزنید تا این فایل برای شما در مسیر Assets/Plugins/Android ساخته شود. همچنین می‌توانید از نمونه موجود در همین مسیر استفاده نمایید که نام آن باید کاملا به صورت proguard-user.txt تغییر پیدا کند.

**وارد کردن اطلاعات محصولات و پرداخت**

اطلاعات محصولات و پرداخت کاملا باید مطابق آن چیزی باشد که در پنل کافه‌بازار ثبت شده است. هرگونه مغایرت یا اشتباه در وارد کردن اطلاعات منجر به عملکرد نادرست در خرید می‌شود. بنابراین برای این مرحله نیاز است که ابتدا بازی خود را در پنل کافه‌بازار آپلود کرده باشید.

**نکته:** برای اینکه بتوانید در پنل کافه‌بازار از بخش پرداخت درون‌برنامه‌ای استفاده کنید، باید ابتدا تنظیمات Manifest را که بالاتر اشاره شد در پروژه انجام دهید، نام پکیج پروژه خود را مشخص کنید و سپس خروجی Apk بگیرید.

بعد از این مراحل، در صحنه خود، “Cafebazaar.ir IAB” را که اضافه کرده‌اید انتخاب کنید. حالا نیاز است در بخش inspector و در “Store Handler” اطلاعات مورد نیاز را وارد کنید.



تصویر ۲: Store Handler در پنجره Inspector

در آرایه Products مقدار size را به تعداد محصولات خود وارد کنید. برای هر محصول Product Id باید کاملا مطابق با شناسه کالا در پنل کافه‌بازار باشد. همچنین نیاز است برای هر محصول مشخص کنید قابل مصرف (Consumable) است یا غیر قابل مصرف (Non Consumable).

برای آشنا شدن با محصولات مصرفی و غیر مصرفی این مقدمه را مطالعه کنید.

https://developers.cafebazaar.ir/fa/docs/iab/intro/

مشخص کردن قیمت محصولات لازم نیست و صرفا در صورت نیاز شخصی می‌توانید وارد نمایید. همچنین می‌توانید در کدهای خود، قیمت محصولات را از سرور کافه‌بازار دریافت کنید که در ادامه نحوه انجام این کار آورده شده است.

بعد از وارد کردن اطلاعات محصولات، مقدار Public Key را از پنل کافه‌بازار کپی کرده و وارد نمایید. این مقدار با نام کلید RSA و در بخش پرداخت درون برنامه‌ای بازی در پنل کافه‌بازار قرار دارد و به ازای هر بازی منحصر بفرد است.

مقدار Payload را به صورت اختیاری و در صورت نیاز می‌توانید هم از طریق Inspector و هم داخل کد وارد نمایید. از Payload برای شناسایی هر درخواست خرید استفاده می‌شود که میتواند هر مقدار دلخواه یا قراردادی باشد. با وارد نکردن آن یا یک مقدار ثابت هیچ اختلالی در فرآیند خرید پیش نخواهد آمد.

**نکته مهم:** شما امکان اجرای فرآیندهای پرداخت درون برنامه‌ای را در محیط یونیتی ندارید و برای تست و خطایابی این کار باید حتما روی دستگاه‌های اندرویدی انجام شود.

در صورتی که میخواهید عملیات بعد از پرداخت و دادن محصول به کاربر را تست و خطایابی کنید می‌توانید گزینه Editor Dummy Response را تیک بزنید. با این کار تمام توابع پرداخت، نتیجه موفقیت آمیز برمی‌گردانند.

**راه‌اندازی سرویس پرداخت درون برنامه‌ای**

برای اینکه بتوانید از پرداخت درون برنامه‌ای و سایر امکانات استفاده کنید، ابتدا باید سرویس کافه‌بازار راه‌اندازی شود. در غیر اینصورت تمام عملیات با شکست مواجه خواهند شد. برای این کار قبل از اقدام کاربر به خرید (در شروع بازی خود یا باز شدن صفحه فروشگاه) به شکل زیر سرویس را راه‌اندازی کنید:

StoreHandler.instance.InitializeBillingService();

همچنین در صورتی که نیاز دارید از نتیجه راه‌اندازی مطلع شوید به شکل زیر عمل کنید:

StoreHandler.instance.InitializeBillingService(OnServiceInitializationFailed, OnServiceInitializedSuccessfully);

نتیجه راه‌اندازی در صورت موفقیت‌آمیز بودن به تابع OnServiceInitializedSuccessfully() و در صورت عدم موفقیت به تابع OnServiceInitializationFailed() ارسال می‌شود که نیاز است به شکل زیر ساخته شوند:

private void OnServiceInitializedSuccessfully()

{

}

private void OnServiceInitializationFailed(int errorCode, string message)

{

}

در تابع OnServiceInitializedSuccessfully() می‌توانید نسبت به انجام سایر کارها مانند درخواست قیمت محصولات از سرور یا بررسی وجود یک محصول در inventory اقدام نمایید.

و در صورت عدم موفقیت در تابع OnServiceInitializationFailed() بر اساس شماره خطا اقدام متناسب انجام دهید. لیست خطاهای احتمالی در انتهای این مطلب آورده شده است.

**فراخوانی درخواست خرید**

می‌توانید با مشاهده اسکریپت InAppStore.cs و صحنه دمو که در مسیر Assets/Bazaar InAppBilling/Demo قرار دارد، درک بهتری از عملکرد پلاگین پیدا کنید.

برای درخواست خرید یک محصول کافی‌ست تابع Purchase از اسکریپت StoreHandler.cs را در هر جای دلخواهی از پروژه به شکل زیر فراخوانی کنید:

StoreHandler.instance.Purchase(index, OnPurchaseFailed, OnPurchasedSuccessfully);

مقدار index شماره محصول در آرایه Products است که از صفر شروع می‌شود.

نتیجه خرید در صورت موفقیت‌آمیز بودن به تابع OnPurchasedSuccessfully() و در صورت عدم موفقیت به تابع OnPurchaseFailed() ارسال می‌شود که نیاز است به شکل زیر ساخته شوند:

private void OnPurchasedSuccessfully(Purchase purchase, int productIndex)

{

}

private void OnPurchaseFailed(int errorCode, string message)

{

}

در تابع OnPurchasedSuccessfully()، بر اساس شماره محصول (productIndex) برای کاربر محصول خریداری شده را فعال یا به موجودی او اضافه کنید. همچنین می‌توانید اطلاعات خرید را از purchase بدست آورید.

و در صورت عدم موفقیت در تابع OnPurchaseFailed() بر اساس شماره خطا اقدام متناسب انجام دهید. لیست خطاهای احتمالی در انتهای این مطلب آورده شده است.

**بررسی وجود یک محصول در Inventory کاربر**

در صورتی که کاربر محصولی را خرید کرده باشد و این محصول غیر مصرفی باشد یا به دلیل خطا مصرف نشده باشد، در Inventory کاربر ذخیره خواهد ماند و کاربر در هر زمان می‌تواند مجددا بدون نیاز به پرداخت هزینه، آن را فعال نماید.

زمانی که می‌خواهید به کاربری که بازی شما را حذف کرده و مجدد نصب کرده، این امکان را بدهید که خریدهای غیر مصرفی خود را بازیابی کند، می‌توانید Inventory کاربر را به شکل زیر بررسی کنید:

StoreHandler.instance.CheckInventory(index, OnInventoryCheckFailed, OnInventoryHadProduct);

مشابه فراخوانی خرید، مقدار index شماره محصول در آرایه Products است که از صفر شروع می‌شود.

نتیجه بررسی در صورت وجود محصول در Inventory کاربر به تابع OnInventoryHadProduct() و در صورت عدم موفقیت به تابع OnInventoryCheckFailed() ارسال می‌شود که نیاز است به شکل زیر ساخته شوند:

private void OnInventoryHadProduct(Purchase purchase, int productIndex)

{

}

private void OnInventoryCheckFailed(int errorCode, string message)

{

}

در تابع OnInventoryHadProduct()، می‌توانید بر اساس شماره محصول (productIndex) برای کاربر محصول موجود در Inventory را فعال کنید. همچنین می‌توانید اطلاعات خرید را از purchase بدست آورید.

و در صورت عدم موفقیت به هر دلیلی (از جمله login نبودن کاربر در کافه‌بازار) در تابع OnInventoryCheckFailed() بر اساس شماره خطا اقدام متناسب انجام دهید. لیست خطاهای احتمالی در انتهای این مطلب آورده شده است.

**درخواست قیمت محصولات**

اگر میخواهید قیمت‌ها را از سرور کافه‌بازار دریافت کنید، بعد از راه‌اندازی موفق سرویس پرداخت، به شکل زیر این کار را انجام دهید:

StoreHandler.instance.LoadProductPrices(OnLoadingPricesFailed, OnPricesLoadedSuccessfully);

نتیجه این عملیات در صورت موفقیت‌آمیز بودن به تابع OnPricesLoadedSuccessfully() و در صورت عدم موفقیت به تابع OnLoadingPricesFailed() ارسال می‌شود که نیاز است به شکل زیر ساخته شوند:

private void OnPricesLoadedSuccessfully()

{

}

private void OnLoadingPricesFailed(int errorCode, string message)

{

}

اجرای تابع OnPricesLoadedSuccessfully() به نشانه این است که قیمت‌ها بارگزاری شده و در آرایه products در Store Handler ذخیره شده است. بنابراین به شکل StoreHandler.instance.products[index].price می‌توانید قیمت هر محصول را نمایش دهید.

**نکته:** قیمت‌ها بر حسب ریال و از نوع string می‌باشند.

**نکته:** ممکن است بعد از تغییر قیمت در پنل کافه‌بازار و اجرای مجدد این تابع، تا مدت کوتاهی همچنان قیمت‌های قبلی برگشت داده شود.

در صورت عدم موفقیت در بارگزاری قیمت‌ها، در تابع OnLoadingPricesFailed() بر اساس شماره خطا اقدام متناسب انجام دهید. لیست خطاهای احتمالی در انتهای این مطلب آورده شده است.

**تایید اعتبار و امنیت پرداخت**

در صورتی که می‌خواهید امنیت خریدهای بازی خود را در برابر کاربران عادی برقرار کنید، می‌توانید قبل از دادن محصول به کاربر، اعتبار هر خرید را توسط API توسعه‌دهندگان تایید کنید. برای این منظور گزینه Validate Purchases را تیک بزنید.

سپس نیاز دارید مطابق راهنمای زیر یک Client در پنل کافه‌بازار بسازید و اطلاعات لازم را وارد نمایید.

**ساخت Client در API توسعه‌دهندگان**

برای این کار در پنل کافه‌بازار به بخش API توسعه‌دهندگان بروید و روی دکمه کلاینت جدید بزنید. در دیالوگ باز شده به جای مقدارRedirect URI آدرس یک وبسایت را وارد کنید. اگر سایت ندارید اهمیتی ندارد، هر آدرس سایتی که وارد کنید قابل قبول است.

دکمه ارسال را بزنید. با این کار یک Client جدید ایجاد می‌شود و تنها نیاز است یک بار انجام شود.

اطلاعات Client Id و Client Secret را کپی کرده و در یونیتی وارد نمایید. اما مقدار Refresh Token را باید با دستورالعمل زیر به درست آورید.

**به دست آوردن Refresh Token**

در مرورگر لینک زیر را کپی کنید:

https://pardakht.cafebazaar.ir/devapi/v2/auth/authorize/?response\_type=code&access\_type=offline&redirect\_uri=XXXXX&client\_id=YYYYY

در این لینک باید دو مقدار را جایگذاری کنید. انتهای لینک به جای XXXXX آدرس Client URI و به جای YYYYY مقدار Client Id از اطلاعات Client را وارد کنید. لینک در نهایت باید بدون فاصله باشد. سپس اینتر بزنید.

در صورتی که لینک درست ایجاد شده باشد، صفحه‌ای باز می‌شود که از شما مجوز دسترسی میخواهد. دکمه تایید را بزنید.

در صورتی که اطلاعات به هر دلیلی اشتباه باشد به شما خطا داده خواهد شد.

بعد از زدن دکمه تایید، آدرسی که در Client وارد کرده‌اید باز خواهد شد و در مرورگر بعد از آدرس، یک پارامتر code دیده می‌شود. به عنوان مثال به شکل زیر:

http://test.com/?code=6hsH0MGCVOzSv1rVMpIGnF

مقدار مقابل code را در مرحله بعدی استفاده می‌کنیم.

سپس لینک زیر را باز کنید.

https://pardakht.cafebazaar.ir/devapi/v2/auth/token/

در صفحه باز شده به جای grant\_type مقدار authorization\_codeرا وارد کنید. به جای code مقدار دریافت شده در مرحله قبل و به جای سه پارامتر دیگر اطلاعات Client ساخته شده را قرار دهید.

سپس روی دکمه Authorize کلیک کنید. در صورتی که اطلاعات درست باشد یک JSON مشابه شکل زیر برگشت داده خواهد شد:

{

"access\_token": "GWObRK06KHLr8pCQzDXJ9hcDdSC3eV",

"token\_type": "Bearer",

"expires\_in": 3600000,

"refresh\_token": "yBC4br1l6OCNWnahJvreOchIZ9B6ze",

"scope": "androidpublisher"

}

شما مقدار refresh\_token را از این JSON نیاز دارید. آن را کپی کرده و در یونیتی وارد نمایید.

با وارد کردن اطلاعات API در یونیتی کار به اتمام می‌رسد و از این پس خریدهای کاربران ابتدا اعتبارسنجی خواهد شد.

**مشکلات متداول در زمان Build و خروجی گرفتن**

۱. در نسخه‌های جدید یونیتی به دلیل استفاده از سیستم Gradle، نیاز به اینترنت و دور زدن تحریم می‌باشد. امکان دارد عدم اتصال به سرورهای Google و Gradle به دلیل تحریم، مانع خروجی گرفتن شود و با خطا روبرو شوید. لطفا قبل از هر چیز با یک پروژه خالی و قبل از اضافه کردن پلاگین، مطمئن شوید در این مرحله اشکالی وجود ندارد و می‌توانید با موفقیت خروجی بگیرید.

۲. در صورت وجود چند پلاگین پرداخت به صورت همزمان در پروژه شما، به دلیل اختلال پلاگین‌ها امکان دارد نتوانید به درستی خروجی بگیرید. برای این کار مطمئن شوید در مسیر Assets/Plugins/Android تنها یک پلاگین پرداخت وجود داشته باشد.

۳. فایل Manifest شما در صورت قدیمی بودن یا عدم سازگاری با نسخه یونیتی شما می‌تواند در زمان خروجی گرفتن مشکل ایجاد کند. همچنین وجود خطاهای دستوری، کاراکترهای اضافی یا عدم رعایت باز و بسته شدن درست تگ‌ها، که باعث شود فایل Manifest قابلیت خوانایی نداشته باشد می‌تواند منجر به مشکل شود.  
برای اطمینان از اینکه فایل Manifest شما مشکلی ندارد، نسخه خام و سازگار با یونیتی خود را می‌توانید از محل نصب یونیتی در سیستم خود، در مسیر Unity/Editor/Data/PlaybackEngines/AndroidPlayer/Apk در پروژه خود کپی کنید. سپس مطابق با نمونه‌های Manifest، دسترسی بازار و اطلاعات Activity را به آن اضافه کنید.

**لیست خطاهای فرآیند پرداخت**

**۱:** مقدار Public RSA Key را وارد نکرده‌اید.

**۲:** کاربر برنامه بازار را بر روی دستگاه خود نصب ندارد.

**۳:** سرویس پرداخت بازار به درستی راه‌اندازی نشده است. می‌توانید مجدد اقدام به راه‌اندازی کنید.

**۴:** خطای داخلی در فرآیند پرداخت و فراخوانی سرویس‌های بازار رخ داده است. لطفا دستورالعمل‌های راه‌اندازی پلاگین را به دقت مطالعه و پیاده‌سازی کنید و در صورت تکرار خطا با پشتیبانی توسعه‌دهندگان کافه‌بازار تماس بگیرید. همچنین لازم به ذکر است که ممکن است این خطا مقطعی یا تحت شرایط خاصی رخ دهد. لطفا بر روی دستگاه‌های دیگر نیز تست نمایید.

**۵:** لغو فرآیند خرید توسط کاربر یا ناموفق بودن پرداخت. این خطا فقط در زمان پرداخت رخ خواهد داد.

**۶:** خرید کاربر به درستی انجام شده است اما فرآیند مصرف کردن محصول با خطا مواجه شده. محصول خریداری شده در Inventory کاربر ذخیره شده و کاربر می‌تواند با اقدام مجدد به خرید، بدون نیاز به پرداخت هزینه فرآیند مصرف کردن را تکرار نماید. احتمال رخ دادن این خطا بسیار کم است و در صورتی رخ خواهد داد که محصول مصرفی باشد.

**۷:** کاربر وارد حساب کاربری خود در برنامه بازار نشده است. این خطا تنها هنگام بررسی Inventory کاربر رخ خواهد داد.

**۸:** این محصول در Inventory کاربر موجود نیست یا مصرف شده است. این خطا تنها هنگام بررسی Inventory کاربر رخ خواهد داد.

**۹:** خطا در فرآیند تایید اعتبار خرید به دلیل مشکلات اینترنت یا اشتباه وارد کردن اطلاعات کلاینت API توسعه‌دهندگان.

**۱۰:** هزینه این خرید به کاربر برگشت داده شده است. این خطا ممکن است پس از بررسی اعتبار خرید رخ دهد.

**۱۱:** استفاده از پلاگین پرداخت تنها بر روی دستگاه‌های اندروید امکان‌پذیر است و نمی‌توانید در محیط یونیتی فرآیند پرداخت را تست و خطایابی کنید.

**۱۲:** شماره محصول در آرایه products موجود نیست. شماره محصولات از صفر شروع می‌شود.

**۱۳:** در آرایه products شناسه محصول نامعتبری را وارد کرده‌اید یا هیچ محصولی در این آرایه وجود ندارد. این خطا هنگام دریافت قیمت‌ها از سرور کافه‌بازار رخ خواهد داد.

**14:** ابتدای درخواست شما هنوز سرویس‌های کافه‌بازار راه‌اندازی نشده بود. اکنون سرویس‌ها راه‌اندازی شده، می‌توانید بدون نمایش پیام خطا به کاربر، درخواست خود را تکرار کنید.