

Configuration for the Amazon and Amazon Underground Stores

Introduction

This guide describes setting up the Amazon Store and the Amazon Underground Store for use with the Unity In-App Purchasing (IAP) system. It covers establishing the digital records and relationships that are required to interact with the Unity IAP API, setting up an Amazon developer account, as well as testing and publishing a Unity IAP application.

Like other platform's stores Amazon allows for the purchase of virtual goods and managed products. These digital products are identified by using a string identifier and an additional type to define durability: *subscription*, *consumable* (capable of being rebought), and *non-consumable* (capable of being bought once).

The Amazon stores are set up to support free and paid versions of the same app. The Amazon Underground store is designed to give developers a schedule for payment for free apps that meet their criteria and the Amazon store is provided to support paid version of apps. The Underground stores have more stringent requirements for apps considering the payment model for developers. These requirements don't allow for developers to actually charge for non-durable purchases, or subscriptions, but follow a pay per time spent in the application. The full details can be found [here](#).

Amazon and Amazon Underground Stores

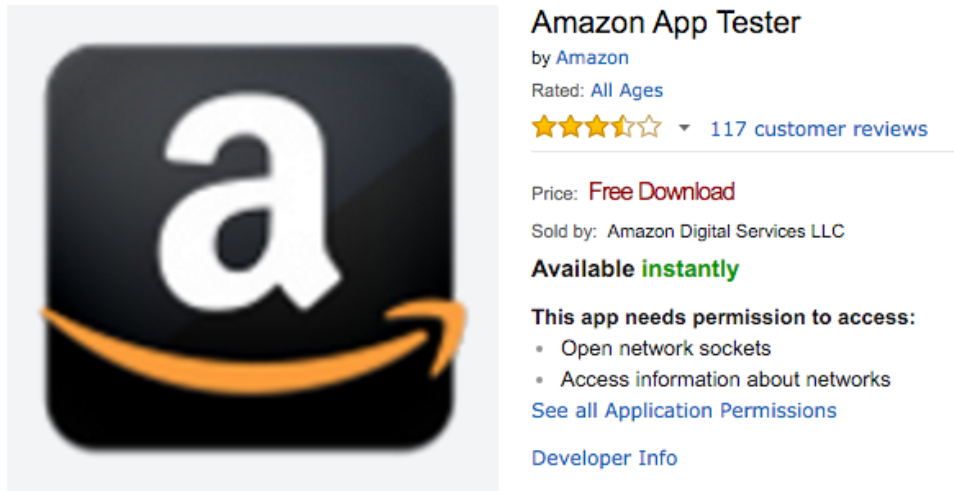
Getting Started

1. Setup an Amazon developer account. Navigate to the [Amazon developer portal](#) and create a developer account. Also note the [differences](#) between the type of apps that are accepted to the Amazon store and the Amazon Underground store. A single developer account will be used to set up both stores.
2. Take note of game's product identifiers and understand the differences in the types of digital items that are acceptable for the different stores. For example, the following identifier for consumable coins will be used in the normal amazon catalog.
3. Write a game implementing the Unity IAP API. For reference, see the [Unity IAP Initialization](#) and [Integrating Unity IAP In Your Game](#) guides. Use the Amazon store for apps with no restrictions on IAP items. For a free version of your app on Amazon Underground make the following [modifications](#).

Device Setup

1. For Android devices download and install the [Amazon App Store](#). Additionally, if you plan on an Underground version of you app download and install the [Amazon Underground App Store](#).
2. On FireOS devices the Amazon store and the Amazon Underground apps should come pre-installed. If your FireOS device doesn't have the Amazon Underground Store installed, you can use the link for the other Android devices that is listed in step one.

3. On both Android and FireOS devices install the [Amazon App Tester](#). Follow the link above or search for this on the Amazon App Store and install it.

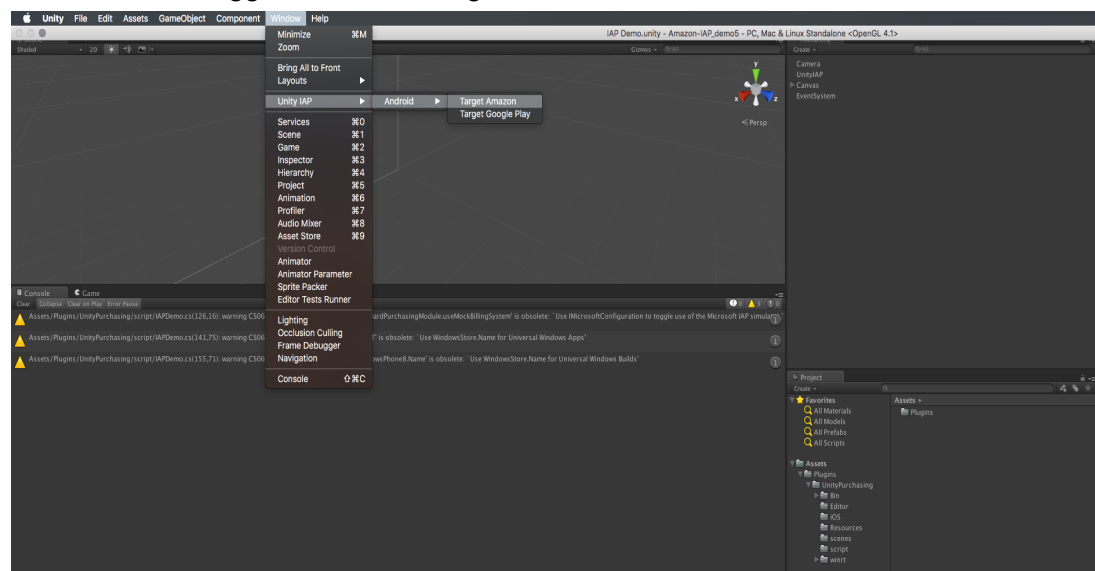


4. Setup the Android tool chain
 - a. Confirm the toolchain recognizes the attached Android device through adb, for example:

```
|[11:07:01] user@laptop:/Applications | $ adb devices
List of devices attached
00DA0807526300W5      device
```

Unity App Setup

1. Using your Unity's IAP can be accomplished in several steps.
 - a. Set the IAP target store. You should already have an Android app set up, set the target store. Use the Window > Unity IAP > Android > Target Amazon menu item. This is used to toggle between Google and Amazon stores.



- i. Alternatively call the API:

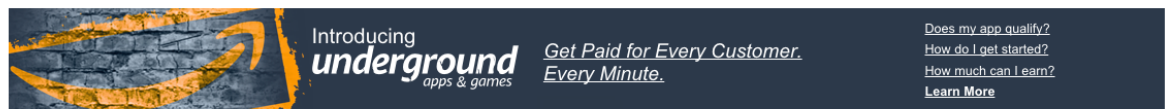
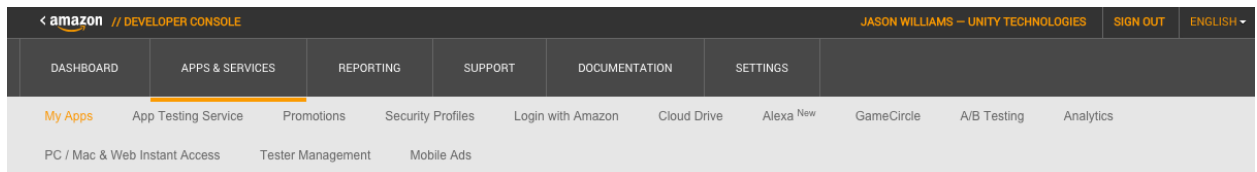
```
UnityPurchasingEditor.TargetAndroidStore(AndroidStore.AmazonAppStore)
```

- b. Note, to fetch the current Amazon User ID for other Amazon services use the `IAmazonExtensions`:

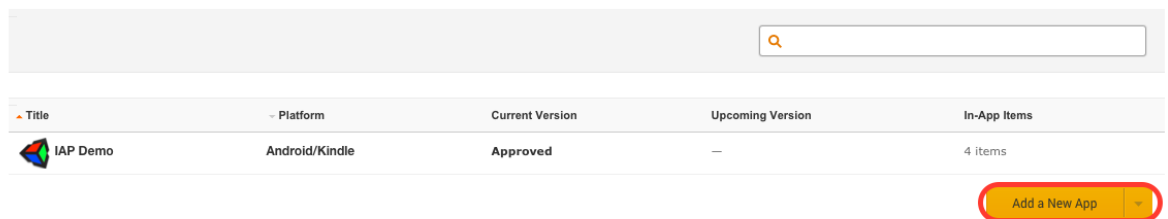
```
public void OnInitialized
    (IStoreController controller, IExtensionProvider extensions)
{
    string amazonUserId =
        extensions.GetExtension<IAmazonExtensions>().amazonUserId;
    // ...
}
```

Amazon Store Setup

1. In preparation for using the Amazon Store, skip download prompts for Amazon's plugins, for example Amazon has several plugins (GameCircle, Amazon's IAP 1.0/2.0 and MobileAds) for Unity that aren't in scope for using the Amazon store. These plugins duplicate functionality that exists in Unity's Engine and service offerings.
2. Add your app. From the Amazon Developer Portal select the "Add a New App".

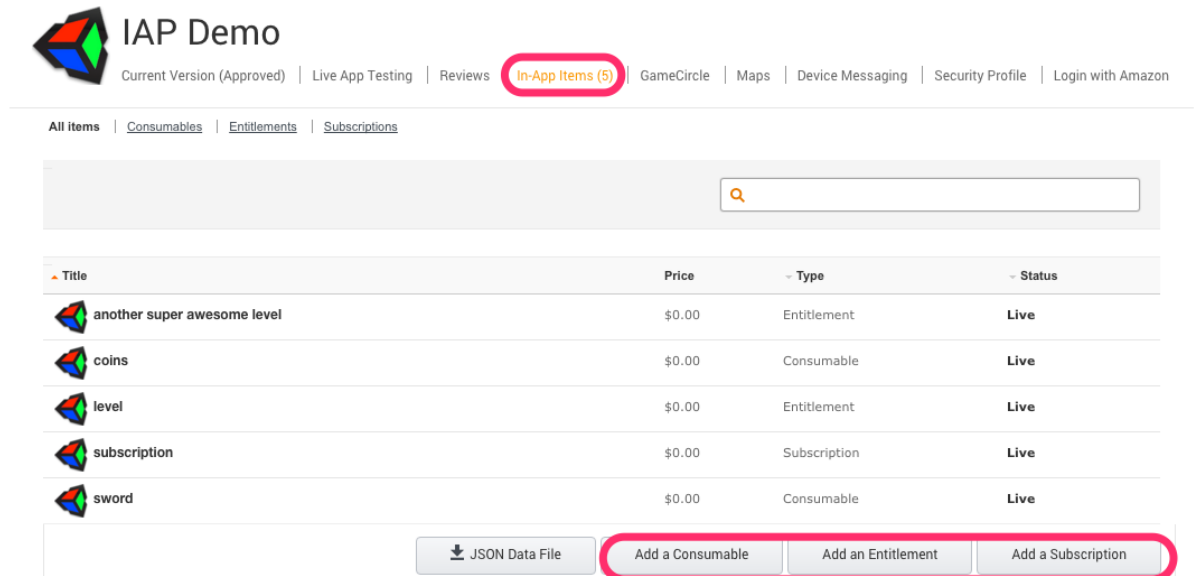


1 apps under this account



3. Setup your test app. Local testing for the Amazon Store requires the app to be installed on the device and the catalog setup in order to generate a JSON file (below). This file will serve to describe your assets.
Installing the APK file to your device can be done using [adb](#) or [Unity](#).
Set up your catalog, using the item descriptions from the getting started, add the items to the Amazon catalog using the Amazon Developer Portal navigate to the app specific

page and find the In-App Items section. Use the “Add a Consumable”, “Add an Entitlement” or “Add a Subscription” buttons to set up your catalog.



The screenshot shows the 'IAP Demo' web interface. At the top, there's a navigation bar with links: 'Current Version (Approved)', 'Live App Testing', 'Reviews', 'In-App Items (5)' (highlighted with a red circle), 'GameCircle', 'Maps', 'Device Messaging', 'Security Profile', and 'Login with Amazon'. Below this is a sub-navigation bar with 'All Items', 'Consumables', 'Entitlements', and 'Subscriptions'. A search bar is present on the right. The main content is a table with the following items:

Title	Price	Type	Status
another super awesome level	\$0.00	Entitlement	Live
coins	\$0.00	Consumable	Live
level	\$0.00	Entitlement	Live
subscription	\$0.00	Subscription	Live
sword	\$0.00	Consumable	Live

At the bottom, there are three buttons: 'JSON Data File' (with a download icon), 'Add a Consumable', 'Add an Entitlement', and 'Add a Subscription' (all three buttons are grouped and highlighted with a red circle).

To use Amazon’s local sandbox testing App generate a JSON description of your product catalog on the device’s SD card using the `IAmazonConfiguration` extended configuration:

```
var builder = ConfigurationBuilder.Instance(  
    StandardPurchasingModule.Instance() );  
// Define your products.  
builder.AddProduct("someConsumable", ProductType.Consumable);  
// Write a product description to the SD card  
// in the appropriate location.  
builder.Configure<IAmazonConfiguration>()  
    .WriteSandboxJSON(builder.products);
```

1. When using this method to write product descriptions to the SD card declare the Android permission to write to external storage in the test app’s manifest:

```
<uses-permission  
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

- a. Remove this extra permission before publishing, if appropriate.
2. The Amazon testing sandbox is now setup for local testing. For more information please see [Amazon's App Tester documentation](#).