Gameramp SDK Documentation

1. Introduction:

The Gameramp SDK is a lightweight Unity SDK that enables analytics, ads tracking, and in-app purchases (IAP) within your game.

Supported Platforms:

- Unity 2021.3+
- Android 7.0+
- iOS 12+

Key Features:

- Analytics (sessions, first-open, custom events)
- Ads tracking (banner, interstitial, rewarded)
- In-App Purchases (IAP)
- Authentication & Player Config
- Store management

2. Quick Start:

Get started with Gameramp SDK in 3 steps:

Step 1: Import the SDK

1. Download the Gameramp SDK package (.unitypackage).
2. In Unity: **Assets** → **Import Package** → **Custom Package.**
3. Select and import the package.
4. Verify `**GamerampSDK/**` folder is present.

Step 2: Create Configuration

1. In the Project window: **Create** → **Gameramp** → **SDKConfig**.
2. Name it GamerampConfig.
3. Fill in credentials in Inspector:
    - SDK Key (required)
    - Secret Key (required)

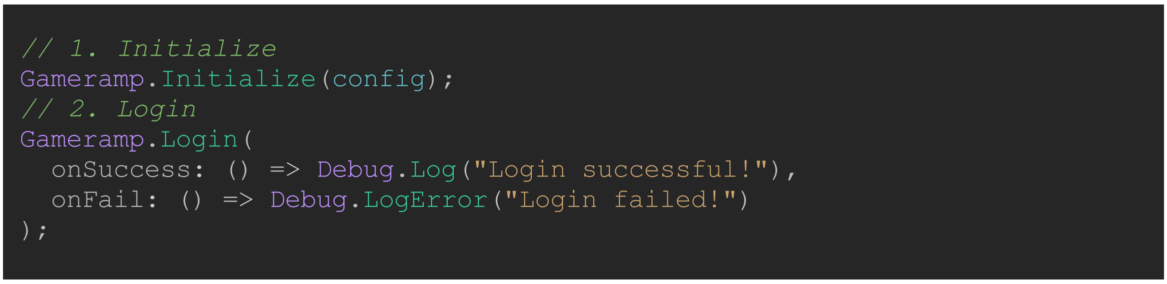
Step 3: Initialize SDK

Option 1 (Recommended): Use the Gameramp Prefab

1. Drag Gameramp.prefab from **GamerampSDK/Prefabs/** into your first scene
2. Assign your **GamerampConfig** to the prefab's config field
3. The prefab will **auto-initialize** and auto-login on Start

Option 2 (Manual): Use GamerampSDK Component

1. Create an empty GameObject
2. Add GamerampSDK.cs component
3. Assign your **GamerampConfig**
4. Call methods below before using any other features from SDK.



```
// 1. Initialize
Gameramp.Initialize(config);
// 2. Login
Gameramp.Login(
  onSuccess: () => Debug.Log("Login successful!"),
  onFail: () => Debug.LogError("Login failed!")
);
```

3. Setup & Configuration:

Android

- Enable custom mainTemplate.gradle in Player Settings.
- Java version: VERSION_11 (Unity 2021.3.45) & VERSION_17 (Unity 6)
- Permissions automatically added: INTERNET, ACCESS_NETWORK_STATE

iOS

- Target minimum iOS 12.0.
- StoreKit automatically included for IAP.

IL2CPP

Unity’s IL2CPP backend strips unused code during builds. To prevent **Gameramp SDK** classes and **Newtonsoft.Json** types from being removed, a link.xml file is required.

- A preconfigured link.xml file is already included with the Gameramp SDK package (**GamerampSDK/link.xml**).
- If your project **already has a link.xml file**:
  - Open it and check whether it contains entries for:
    - GamerampCoreDLL / GamerampUnitySDK
    - Newtonsoft.Json
  - If these entries are missing, copy them from the provided file and add them into your existing link.xml.
- If your project does **not** have a link.xml:
  - Move or copy the provided file into your project’s **Assets/** root directory.

Note: Unity only respects link.xml files inside the Assets/ directory. Files left inside the SDK package folder will be ignored during build.

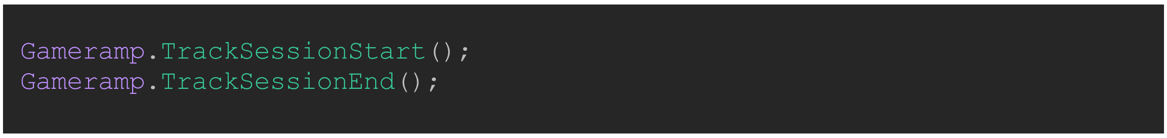
4. Feature Guides:

4.1 Analytics

Track user sessions and app events.

Session Tracking

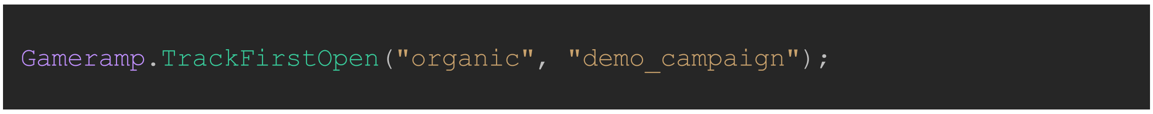
Example:



```
Gameramp.TrackSessionStart();
Gameramp.TrackSessionEnd();
```

First Open Tracking

Example:

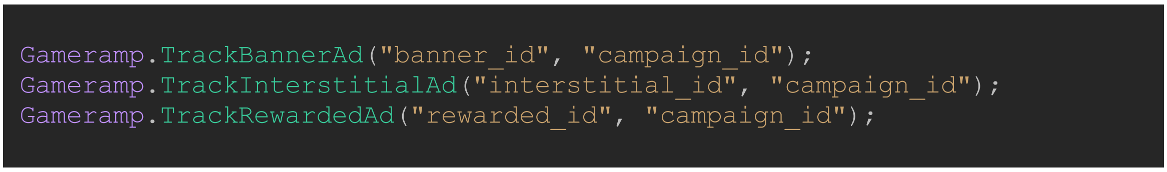


```
Gameramp.TrackFirstOpen("organic", "demo_campaign");
```

4.2 Ads:

Track ad impressions for banners, interstitial, and rewarded ads.

Examples:



```
Gameramp.TrackBannerAd("banner_id", "campaign_id");
Gameramp.TrackInterstitialAd("interstitial_id", "campaign_id");
Gameramp.TrackRewardedAd("rewarded_id", "campaign_id");
```

4.3 In-App Purchases (IAP):

Implement in-app purchases with Gameramp SDK.

Fetch Store Products:

Example:

```
// Fetch store products
Gameramp.FetchStoreProducts(
   (products) => {
       foreach (var sku in products.activeSkus) {
           Debug.Log($"{sku.name} ({sku.skuId}) - {sku.price}");
       }
   },
   (error) => Debug.LogError($"Failed to fetch products: {error}")
);
```

Initiate Purchase:

Example:

```
Gameramp.InitiatePurchase("sku_id",
   onSuccess: (purchase) => {
       Debug.Log($"Demo: Purchase successful via callback - 
{purchase.productId}");
   },
   onFailed: (error) => {
       Debug.LogError($"Demo: Purchase failed via callback - {error}");
   }
);
```

Events:

Subscribe to purchase events:

```
Gameramp.OnPurchaseCompleted += (purchase) => { ... };
Gameramp.OnPurchaseFailed += (error) => { ... };
```

5. API Reference:

Core Methods:

Method	Description	Parameters	Returns
Initialize	Initialize SDK	config – SDK config	void
Login	Authenticate user	onSuccess, onFail	void
Cleanup	Release SDK resources	–	void
Logout	Logout current user	–	void

Ads:

Method	Description	Parameters	Returns
TrackBannerAd	Track banner ad impression	adId, campaignId	void

TrackInterstitialAd	Track interstitial ad impression	adId, campaignId	void
TrackRewardedAd	Track rewarded ad impression	adId, campaignId	void

Analytics Methods:

Method	Description	Parameters	Returns
TrackSessionStart	Track app session start	–	void
TrackSessionEnd	Track app session end	–	void
TrackFirstOpen	Track first app open	source, campaign	void

IAP Methods:

Method	Description	Parameters	Returns
FetchStoreProducts	Fetch available store products	onSuccess, onError callbacks	void
InitiatePurchase	Start purchase flow	productId, onSuccess, onFailed	void
ReinitializeIAP	Reinitialize IAP after resume	–	void

Note: All tracking methods support optional success/error callbacks for advanced tracking.

6. Troubleshooting & FAQ:

- SDK not initializing? → Check if config keys are assigned.
- IL2CPP build errors? → Ensure link.xml file is present.
- No logs in Unity console? → Enable GRLogger.IsLoggingEnabled = true.
- IAP not available? → Verify StoreKit (iOS) / Google Play (Android) setup.