Superfine SDK Unity

1 Setup

1.1 Import Unity package

Download the SuperfineSDK zip file, unzip it, and copy the extracted files to your Packages folder.

* The SDK requires JSON.NET and external-dependency-manager to function properly. We have included them within the SDK, but you have the flexibility to remove or modify them to match the version requirements of your application from the file: **Packages/com.superfine.attribution/package.json**

```
"com.unity.nuget.newtonsoft-json": "3.2.1",
"com.google.external-dependency-manager": "1.2.176"
```

1.2 Get App Information

Go to the project section on the Superfine.org dashboard, select the project, and copy the **Project ID** and **Superfine App Secret**

1.3 Update Superfine Setting

From the menu pick **Superfine/Edit Settings** Update your **Project ID**, **Superfine App Secret**

1.4 Initialize SDK

Add code to initialize the SDK (could be placed in the Awake function of a new component).

```
void Awake()
{
    SuperfineSDKInitOptions options = new SuperfineSDKInitOptions();

#if !UNITY_EDITOR
#if UNITY_ANDROID
    // Enable VERBOSE (or INFO, DEBUG) logging for Android.
    options.logLevel = LogLevel.VERBOSE;
#elif UNITY_IOS
    // Enable debug mode for iOS
    options.debug = true;
#endif
#endif
       // Create an instance of SuperfineSDK
    SuperfineSDK.CreateInstance(options);
}
```

SuperfineSDKInitOptions: Can't be null. Contains:

- flushInterval (**Long**): Time interval (milliseconds) for data flush to the server.
- flushQueueSize (**Integer**): Maximum number of stored events before server flush.
- customUserId (**Boolean**): Flag for using a custom user ID.
- userId (**String**): Custom user identifier to associate events with a user
- waitConfigId (**Boolean**): Flag to wait for the configuration ID before starting.
- autoStart (**Boolean**). Flag to automatically start the SDK on initialization.
- storeType (**StoreType**): Can be UNKNOWN, GOOGLE_PLAY, APP_STORE
- logLevel (**LogLevel**): for Android only. Can be VERBOSE, INFO, DEBUG)
- debug (**Boolean**): for iOS only. If set to true, will display the debug log
- captureInAppPurchases (**Boolean**): for iOS only. Enable capturing in-app purchases for iOS

You can start it whenever you like if you set autoStart = false

```
options.autoStart = false;
SuperfineSDK.CreateInstance(options);

//Start SuperfineSDK when you want to
SuperfineSDK.Start();
```

You can stop SuperfineSDK by calling this function:

```
SuperfineSDK.Stop();
```

2 Send Events

2.1 Wallet Events

LogWalletLink

`void LogWalletLink(const String wallet, const String type = "ethereum");`

Call this method when you want to link the user's wallet address.

Parameters	
`wallet`	**String**: Can't be null. The wallet address that you want to log the linking event for
`type`	**String**: Default is "ethereum". The chain of the wallet address

Example:

```
//Link wallet address "ronin:50460c4cd74094cd591455cad457e99c4ab8be0" in the "ronin" c
SuperfineSDK.LogWalletLink("ronin:50460c4cd74094cd591455cad457e99c4ab8be0", "ronin");
```

LogWalletUnlink

`void LogWalletUnlink(const String& wallet, const String& type = "ethereum");`

Call this method when you want to unlink the user's wallet address.

Parameters	
`wallet`	**String**: Can't be null. The wallet address that you want to log the linking event for
`type`	**String**: Default is ethereum. The chain of the wallet address

Example:

```
//Unlink wallet address "ronin:50460c4cd74094cd591455cad457e99c4ab8be0" in "ronin" cha
SuperfineSDK.LogWalletUnlink("ronin:50460c4cd74094cd591455cad457e99c4ab8be0", "ronin")
```

2.2 Game Level Events

LogLevelStart

`void LogLevelStart(int id, const String name);`

Call this method when starting a level:

Parameters	
`id`	**Integer**: Can't be null. The level id that you want to log.
`name`	**String**: Can't be null. The name of the level that you want to log.

Example:

```
// Log start level ID 10 with the name "level_10"
SuperfineSDK.LogLevelStart(10,"level_10");
```

LogLevelEnd

`void LogLevelEnd(int32 id, const String name, bool isSuccess);`

Call this method when completing a level.

Parameters	
`id`	**Integer**: Can't be null. The level id that you want to log.
`name`	**String**: Can't be null. The name of the level that you want to log.
`isSuccess`	**Boolean**: Can't be null. True if you pass the level. False if you can't pass the level.

Example:

```
//Log you completed level ID 10 with the name "level_10" and you won.
SuperfineSDK.LogLevelEnd(10, "level_10", true);
```

2.3 Ads Events

These events are used to track ads from your app. You can use the Superfine dashboard later to check ad performance based on these events.

LogAdLoad

`void LogAdLoad(const String adUnit, AdPlacementType adPlacementType, AdPlacement adPlacement = AdPlacement::UNKNOWN);`

Call this method when an ad placement is loaded.

Parameters	
`adUnit`	**String**: Can't be null. The ad unit that you want to log.
`adPlacementType`	**enum AdPlacementType**: Can't be null. Can be: - BANNER = 0 - INTERSTITIAL = 1 - REWARDED_VIDEO = 2 - adPlacement

Parameters	
`adPlacement`	**enum AdPlacement**: Can't be null. Can be: - UNKNOWN = -1 - BOTTOM = 0 - TOP = 1 - LEFT = 2 - RIGHT = 3 - FULL_SCREEN = 4

Example:

```
//Log ad unit "ad_unit_test" with ad placement type is "INTERSTITIAL" at placement "FU
SuperfineSDK.LogAdLoad("ad_unit_test", AdPlacementType.INTERSTITIAL, AdPlacement.FULL_
```

LogAdClosed

`void LogAdClosed(string adUnit, AdPlacementType adPlacementType, AdPlacement adPlacement = AdPlacement.UNKNOWN);`

Call this method when an ad is closed.

Parameters	
`adUnit`	**String**: Can't be null. The ad unit that you want to log.
`adPlacementType`	**enum AdPlacementType**: Can't be null. Can be: - BANNER = 0 - INTERSTITIAL = 1 - REWARDED_VIDEO = 2 - adPlacement
`adPlacement`	**enum AdPlacement**: Can't be null. Can be: - UNKNOWN = -1 - BOTTOM = 0 - TOP = 1 - LEFT = 2 - RIGHT = 3 - FULL_SCREEN = 4

Example:

```
//Log ad unit "ad_unit_test" with ad placement type is "INTERSTITIAL" at placement "FU
SuperfineSDK.LogAdClosed("ad_unit_test", AdPlacementType.INTERSTITIAL, AdPlacement.FUL
```

LogAdImpression

`void LogAdImpression(string adUnit, AdPlacementType adPlacementType, AdPlacement adPlacement = AdPlacement.UNKNOWN);`

Call this method when the ad impression is displayed.

Parameters	
`adUnit`	**String**: Can't be null. The ad unit that you want to log.
`adPlacementType`	**enum AdPlacementType**: Can't be null. Can be: - BANNER = 0 - INTERSTITIAL = 1 - REWARDED_VIDEO = 2 - adPlacement
`adPlacement`	**enum AdPlacement**: Can't be null. Can be: - UNKNOWN = -1 - BOTTOM = 0 - TOP = 1 - LEFT = 2 - RIGHT = 3 - FULL_SCREEN = 4

Example:

```
//Log ad unit "ad_unit_test" with ad placement type is "INTERSTITIAL" at placement "FU
SuperfineSDK.LogAdImpression("ad_unit_test", AdPlacementType.INTERSTITIAL, AdPlacement
```

LogAdClick

`void LogAdClick(string adUnit, AdPlacementType adPlacementType, AdPlacement adPlacement = AdPlacement.UNKNOWN);`

Call this method when the user clicks on an ad.

Parameters	
`adUnit`	**String**: Can't be null. The ad unit that you want to log.
`adPlacementType`	**enum AdPlacementType**: Can't be null. Can be: - BANNER = 0 - INTERSTITIAL = 1 - REWARDED_VIDEO = 2 - adPlacement

Parameters	
`adPlacement`	**enum AdPlacement**: Can't be null. Can be: - UNKNOWN = -1 - BOTTOM = 0 - TOP = 1 - LEFT = 2 - RIGHT = 3 - FULL_SCREEN = 4

Example:

```
//Log ad unit "ad_unit_test" with ad placement type is "INTERSTITIAL" at placement "FU
SuperfineSDK.LogAdClick("ad_unit_test", AdPlacementType.INTERSTITIAL, AdPlacement.FULL
```

LogAdRevenue

`void LogAdRevenue(string network, double revenue, string currency, string mediation = "", SimpleJSON.JSONObject networkData = null)`

Call this method to record revenue obtained from an advertisement.

Parameters	
`network`	**String**: Can't be null. The ad-network that generated the revenue.
`revenue`	**double**: Can't be null. The amount of revenue obtained from the ad-network.
`currency`	**String**: The currency code (e.g., "USD", "EUR") corresponding to the revenue.
`mediation`	**String**: Can be empty. The mediation platform used (e.g., Direct, Max, LevelPlay, etc.).
`networkData`	**SimpleJSON.JSONObject**: Can be null. Additional information about the ad-network.

Example:

```
SuperfineSDK.LogAdRevenue(testAdNetworkId, 1.0, "USD", "DIRECT", new Superfine.Unity.S
        {
            { "param1", "abc" },
            { "param2", 100 },
            { "param3", true }
        }
);
```

2.4 IAP Events

These events are used to track in-app purchases from your app.

LogIAPResult

`void LogIAPResult(string pack, float price, int amount, string currency, bool isSuccess);`

Call this method when the user attempts to buy an IAP item.

Parameters	
`pack`	**String**: Can't be null. The unique identifier of the purchased item or pack.
`price`	**Float**: Can't be null. The price of the item or pack.
`amount`	**Integer**: Can't be null. The quantity of the purchased item or pack.
`currency`	**String**: The currency code (e.g., "USD", "EUR") corresponding to the revenue.
`isSuccess`	**Boolean**: Can't be null. True if the purchase was successful. False if the purchase failed.

Example:

```
//Log when the user completed to purchase a package with ID "test_pack" for 0.99 USD,
SuperfineSDK.LogIAPResult("test_pack", 0.99, 150, "USD", true);
```

LogAPRestorePurchase

`void LogAPRestorePurchase();`

Call this method when the user attempts to buy an IAP item.

Example:

```
SuperfineSDK.LogIAPRestorePurchase();
```

2.5 Custom Events

Log

`void Log(string eventName, int data);`

Call this method to log a custom event with integer data.

Parameters	
`eventName`	**String**: Can’t be null. The name of your custom event.
`data`	**Int**: Can’t be null. The integer data associated with the event.

`void Log(string eventName, string data);`

Call this method to log a custom event with string data.

Parameters	
`eventName`	**String**: Can’t be null. The name of your custom event.
`data`	**String**: Can’t be null. The string data associated with the event.

`void Log(string eventName, Dictionary<string, string> data = null);`

Call this method to log a custom event with dictionary data.

Parameters	
`eventName`	**String**: Can’t be null. The name of your custom event.
`data`	**Dictionary<string, string>**: Default is null. The dictionary data associated with the event.

`void Log(string eventName, SimpleJSON.JSONObject data = null);`

Call this method to log a custom event with JSON object data.

Parameters	
`eventName`	**String**: Can’t be null. The name of your custom event.
`data`	**SimpleJSON.JSONObject**: Default is null. The JSON object data associated with the event.

Example:

```
// Log a custom event with JSON object data
SimpleJSON.JSONObject eventData = new SimpleJSON.JSONObject();
eventData["score"] = 1000;
eventData["level"] = "beginner";
//Log your event
SuperfineSDK.Log("My_Custom_Event_Name", eventData);
```

2.6 Addons

Ads reporting helper class

We offer ad revenue reporting support through our addon classes. Automatically receive detailed reports by implementing the appropriate class based on your chosen mediation platform and registering for events. Currently, we provide support for the following mediations: Max Mediation (AppLovin), IronSource Mediation, and Google AdMob Mediation.

Applovin Addons Helper Class

- **Integration**: Add the AppLovin Helper Addons to your project by going to the **Superfine** > **Copy AppLovin Addon**.
- **Event Registration**: Begin logging revenue and impressions by calling `SuperfineSDKApplovin.RegisterPaidEvent()` . When you're done, turn it off with `SuperfineSDKApplovin.UnregisterPaidEvent()` or when your manager class is destroyed.

IronSource Addons Helper Class

- **Integration**: Add the Ironsource Helper Addons to your project by going to the **Superfine** > **Copy IronSource Addon**.
- **Event Registration**: Begin logging revenue and impressions with `SuperfineSDKIronSource.RegisterPaidEvent()` . Turn it off with `SuperfineSDKIronSource.UnregisterPaidEvent()` when done or when your manager class is removed.

Google AdMob Addon Helper Class

- **Integration**: Add the Admob Helper Addons to your project by going to the **Superfine** > **Copy Admob Addon**. -**Event Registration**: For Google Admob you have to register events for all placement that you have.
    - For Banner: `SuperfineSDKAdMob.RegisterBannerViewPaidEvent(bannerView, adUnitId)` and `SuperfineSDKAdMob.UnregisterBannerViewPaidEvent(bannerView, adUnitId)` .
    - For Interstitial: `SuperfineSDKAdMob.RegisterInterstitialAdPaidEvent(interstitialAd, adUnitId)` and `SuperfineSDKAdMob.UnregisterInterstitialAdPaidEvent(interstitialAd, adUnitId)` .
    - For Rewarded video ad: `SuperfineSDKAdMob.RegisterRewardedAdPaidEvent(rewardedAd, adUnitId)` and `SuperfineSDKAdMob.UnregisterRewardedAdPaidEvent(rewardedAd, adUnitId)` .
    - For Rewarded Interstitial Ads `SuperfineSDKAdMob.RegisterRewardedInterstitialAdPaidEvent(rewardedInterstitialAd, adUnitId)` and `SuperfineSDKAdMob.UnregisterRewardedInterstitialAdPaidEvent(rewardedInterstitialAd, adUnitId)` .

- For App Open `SuperfineSDKAdMob.RegisterAppOpenAdPaidEvent(appOpenAd, adUnitId)` and `SuperfineSDKAdMob.UnregisterAppOpenAdPaidEvent(appOpenAd, adUnitId)` .
- For the Ad revenue `SuperfineSDK.LogAdRevenue()` .

Facebook Events Helper Class

The SuperfineSDKFacebook class simplifies sending events to Facebook for marketing purposes. To smoothly integrate this feature, follow these steps:

- **Integration**: Add the Facebook Addon by copying it from the Superfine menu.
- **Event Registration**: Begin logging events with `SuperfineSDKFacebook.RegisterSendEvent()` . When done, turn it off using `SuperfineSDKFacebook.UnregisterSendEvent()` or when your manager class is removed.

By following these instructions, you can effectively utilize the SuperfineSDKFacebook class to transmit custom events to Facebook, enhancing marketing insights and decision-making for your app.

2.7 Postback Conversion Value for iOS

The method allows you to update both the conversion value and coarse conversion values, and it provides the option to send the postback before the conversion window ends. Additionally, it allows you to specify a completion handler to handle situations where the update fails.

```
using Superfine.Tracking.Unity;

// Example 1: Basic usage without coarseValue or lockWindow
SuperfineSDK.UpdatePostbackConversionValue(10);

// Example 2: Usage with coarseValue
SuperfineSDK.UpdatePostbackConversionValue(8, "low");

// Example 3: Usage with lockWindow
SuperfineSDK.UpdatePostbackConversionValue(15, "medium", true);
```