

Unity IAP for Xiaomi

1. Overview

1.1 What is Unity IAP for Xiaomi IAP

Unity IAP provides an easy way to integrate Xiaomi IAP with Unity.

1.2 Process of using Unity IAP

Please append “.mi” to your package name, e.g. “com.unity.mygame.mi”

a. Technical Integration

- I. Onboarding to Unity
- II. Server side integration
- III. Client side integration

b. Build and Submit APKs

2. Login & Purchase Flow

Unity IAP both supports the stand-alone game (without a game server) and online game (with a game server).

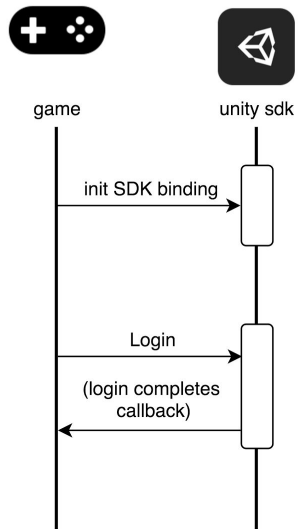
2.1 Stand-alone login & purchase

In the ‘init SDK binding’ step, the Xiaomi appId, Xiaomi appKey (both of these given by Xiaomi) and the Unity Client Id, Unity Client Key (both of these given by Unity, see 2.1 Onboarding to Unity for details) will be used to initialize the Unity IAP SDK.

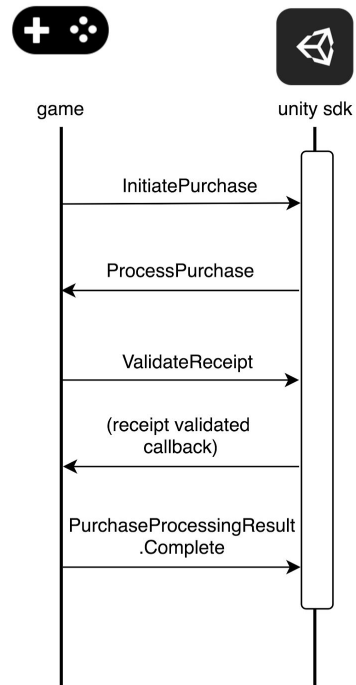
Since there is no game servers, the purchase should be validated on the client side. Sample of the validation can be found in 2.3.5 Purchase Validation. After the validation succeeds, the purchase can be completed.

The flow’s details are shown as below.

Login



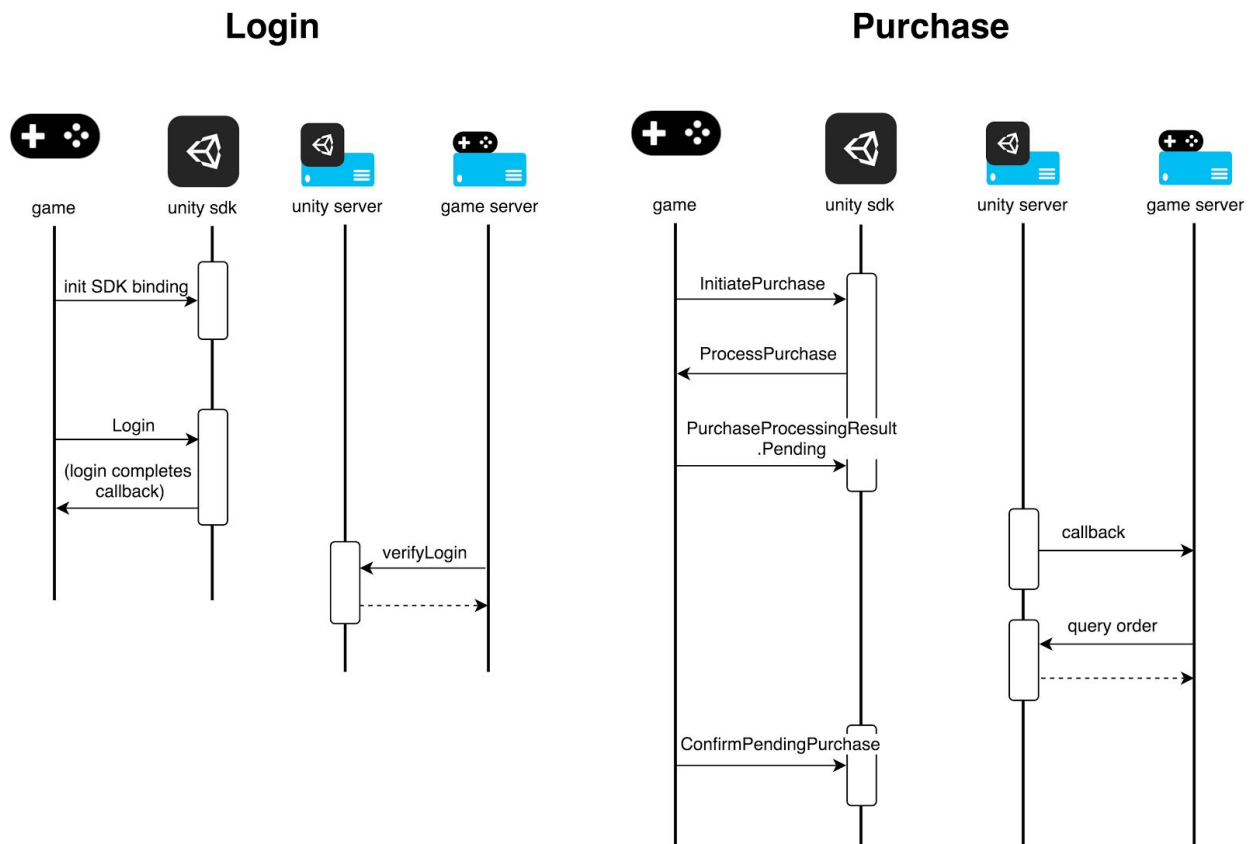
Purchase



2.1 Online login & purchase

Different from stand-alone flows, the game server can verify the login and receive purchase order callback or initiatively query purchase order to/from the Unity IAP server (the details of these APIs are shown in 2.2 Server side integration).

The flow's details are shown as below.



2. Technical Integration

2.1 Onboarding to Unity

Before integration can start, Game developer first needs to onboard to Unity IAP and will obtain the following developer credentials (these keys are binding to the project and bundle id):

Unity Client Id

Unity Client Key

Unity Client Secret

Unity RSA public key

Meanwhile, Game needs to provide the appId, appKey and appSecret offered by Xiaomi, and a callback URL of the Game server (if there exists one) to receive order status update.

Before going live, Game developer will also need to provide Game server IP addresses to be whitelisted on Unity Channel's production server.

2.2 Server side integration

Product environment: <https://cn-api.unity.com>

2.2.1 Verify Xiaomi Logins

GET

/v1/login-attempts/verifyLogin?

userLoginToken=<userLoginToken>&sign=<sign>

Parameters:

Attribute Name	Required?	Description	format	Sample
userLoginToken	Yes	Unity user login token returned by client SDK when finishing login	String	eyJleHRlc5hbEFw cElkljogljQwMzgwN TU0MzkiLCAiZXh0Z XJuYWxUeXBlljogIk ZBS0VDSEwiLCAiZ Xh0ZXJuYWxVaW QiOiAiRkFLRUNITF 9seXhoaylsICJleHR lcm5hbFNlc3Npb24i OiAiRkFLRUNITF9T RVNTSU9OIj0=
sign	Yes	md5 with userLoginToken and Unity Client Secret. Check the algorithm in the end of the doc	String	dc03e329e44c80b 73d5bbfd30207de 8f

Response: in Json

Attribute Name	Required?	Description	Format	Sample
success	Yes	result of verify Xiaomi login	bool	true
errMsg	Yes	the error message of the login if login failed	String	Invalid appld error

Sample:

Request:

GET

/v1/login-attempts/verifyLogin?

userLoginToken=eyJleHRlcm5hbEFwcElkljogIjQwMzgwNTU0MzkiLCAiZXh0ZXJuYWxUeXBlljogIklZBS0VDSEwiLCAiZXh0ZXJuYWxVaWQiOiAiRkFLRUNITF9seXhoayIsICJleHRlcm5hbFNlc3Npb24iOiAiRkFLRUNITF9TRVNTSU9OIj0=&sign=ea24156d9c6fa3d527e99b9d644068ab

Response:

```
{
  "success": true
}
```

2.2.2 Check Order Status

GET

/v1/order-attempts/query?cpOrderId=<cpOrderId>&clientId=<clientId>&orderQueryToken=<orderQueryToken>&sign=<sign>

Parameters:

Attribute Name	Required ?	Description	Format	Sample
cpOrderId	Yes	order id assigned by game , or Unity if game will not generate it	String	66mea52wne
clientId	Yes	client id assigned by Unity after onboarding	String	Q4AnJDW2-r xLAPujqrk1z Q
orderQueryToken	Yes	order query token returned by client SDK when finishing purchase	String	XIAOMI
sign	Yes	md5 with cpOrderId, clientId, orderQueryToken and Unity Client Secret. Check the algorithm in the end of the doc	String	7277da780d1 102864ee581 8c2730c74e

Response: in Json

Attribute Name	Required?	Description	Format	Sample
id	Yes	orderAttempt id assigned by Unity	String	27487794382 6

clientId	Yes	client id returned after Game onboarding to Unity Channel	String	Q4AnJDW2-r xLAPujqrk1z Q
cpOrderId	Yes	order id assigned by game , or Unity if game will not generate it	String	66mea52wne
uid	Yes	user id assigned by Xiaomi	String	15400813498
status	Yes	status of the orderAttempt	String	SUCCESS
productId	Yes	product id of the product associated with the orderAttempt	String	Product_1
payFee	Yes	pay fee of this order (in channel unit)	Integer	1
quantity	Yes	quantity of the product	Integer	1
notified	Yes	whether the orderAttempt's status is notified to Game server	Boolean	No
currency	Yes	currency	ISO 4217	CNY
country	Yes	country	ISO 3166-2	CN
paidTime	Yes	orderAttempt paid time	ISO8601 yyyy-MM-dd Thh:mm:ssX XX, UTC timezone	2017-03-08T 06:43:20Z
createdTime	Yes	orderAttempt created time	ISO8601 yyyy-MM-dd Thh:mm:ssX	2017-03-08T 06:43:20Z

			XX, UTC timezone	
updatedAt	Yes	orderAttempt updated time	ISO8601 yyyy-MM-dd Thh:mm:ssX XX, UTC timezone	2017-03-08T 06:43:20Z
createdBy	Yes	created by which user (only for support use)	String	27487794382 2
updatedBy	Yes	updated by which user (only for support use)	String	0
rev	Yes	the revision of the orderAttempt (only for update)	String	0
extension	No	extesion	Json String	{"abc" : "123"}

Sample:

Request:

GET

/v1/order-attempts/query?cpOrderId=<66mea52wne>&clientId=<Q4AnJDW2-rxLAPujqrk1zQ>&orderQueryToken=<eyJjaGFubmVsVHlwZSI6ICJGQUtFQ0hMliwglmNoYW5uZWxVaWQiOiAiRkFLRUNITF9oeGQwNSJ9>&sign=<debd0018911d263e23dfb729207b905c>

Response:

```
{
  "createdBy" : "0",
  "updatedBy" : "0",
  "createdTime" : "2017-03-08T06:43:20Z",
  "updatedAt" : "2017-03-08T06:43:20Z",
```



```

    "id" : "274877943826",
    "userId" : "274877943822",
    "status" : "SUCCESS",
    "cpOrderId" : "66mea52wne",
    "clientId" : "Q4AnJDW2-rxLAPujqrk1zQ",
    "uid" : "15400813498",
    "productId" : "Product_1",
    "payFee" : 1,
    "quantity" : 1,
    "paidTime" : "2017-03-08T06:43:20Z",
    "notified" : false,
    "currency" : "CNY",
    "country" : "CN",
    "rev" : "1"
}

```

2.2.3 OrderAttempt Callback Notification

Unity Channel server will notify the Game server with the order payment result. Please implement the http GET and accept following query parameters.

Attribute Name	Required?	Description	Format
signData	Yes	the order content which is in Json String format (details can be seen in the table below)	Json String
signature	Yes	RSA signature with the signData. Check the algorithm in the end of the doc	String

signData content:

Attribute Name	Required?	Description	Format	Sample
orderAttemptId	Yes	orderAttempt id assigned by Unity	String	274877943826
status	Yes	status of the orderAttempt	String	SUCCESS
productId	Yes	product id of the product associated with the orderAttempt	String	Product_1
quantity	Yes	quantity of the product	Integer	1
extension	No	extesion	Json String	{"abc" : "123"}
paidTime	Yes	orderAttempt paid time	yyyy-MM-ddT hh:mm:ssXXX , GMT+8 timezone	2017-03-08 06:43:20
cpOrderId	Yes	order id assigned by Game	String	66mea52wne
clientId	Yes	client id returned after Game onboarding to Unity Channel	String	P-wU0n9pbyQ1ulks4kHX_Q
country	Yes	currency	String of ISO 4217	CNY
currency	Yes	country	String of ISO 3166-2	CN
payFee	Yes	pay fee of this order (in channel unit)	String	1

Response:

Please return "ok". Or there will be retry notification in a time range.

Enum types:

1. status: SUCCESS, FAILED, UNCONFIRMED

Signature verification:

1. Api call:

- a. Any api call to Unity Channel server should have the parameter of signature.
- b. Use all the parameters to calculate the sign if they are not null or empty.
- c. The parameters should be urlDecoded.
- d. Sort the parameters by alphabet of the attribute names, and then concatenate the attribute values with the delimiter of &, finally append the app secret at the end of the string to be signed
- e. Do not use the sign in the sign string.
- f. Use md5 to hash the string. Use the lower case result to verify sign

Example:

When there is a call look like:

GET

/v1/login-attempts/verifyLogin?

userLoginToken=eyJleHRlcm5hbEFwcElkljogIjQwMzgwNTU0MzkiLCAiZXh0ZXJuYWxUeXBlljogIkZBS0VDSEwiLCAiZXh0ZXJuYWxVaWQiOiAiRkFLRUNITF9seXhoayIsICJleHRlcm5hbFNlc3Npb24iOiAiRkFLRUNITF9TRVNTSU9OIj0=&sign=ea24156d9c6fa3d527e99b9d644068ab

Then the string to be signed is

eyJleHRlcm5hbEFwcElkljogIjQwMzgwNTU0MzkiLCAiZXh0ZXJuYWxUeXBlljogIkZBS0VDSEwiLCAiZXh0ZXJuYWxVaWQiOiAiRkFLRUNITF9seXhoayIsICJleHRlcm5hbFNlc3Npb24iOiAiRkFLRUNITF9TRVNTSU9OIj0=&{SECRET}

where {SECRET} is the *Unity Client Secret* returned after onboarding.

2. Callback:

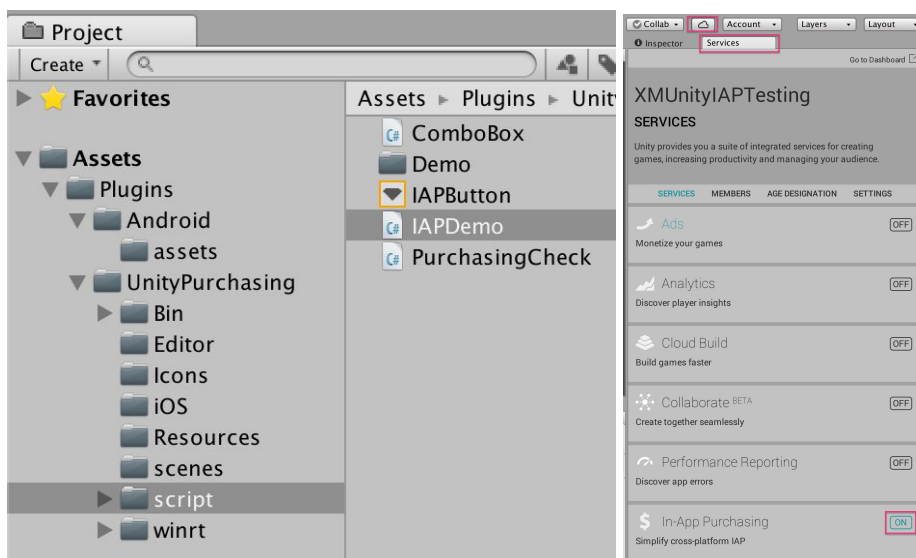
When Unity Channel server gives the callback notification to Game server, it will use RSA signature for verification. Please use the Unity Channel RSA public key returned after onboarding to verify the *signData* with the *signature*.

2.3 Client side integration

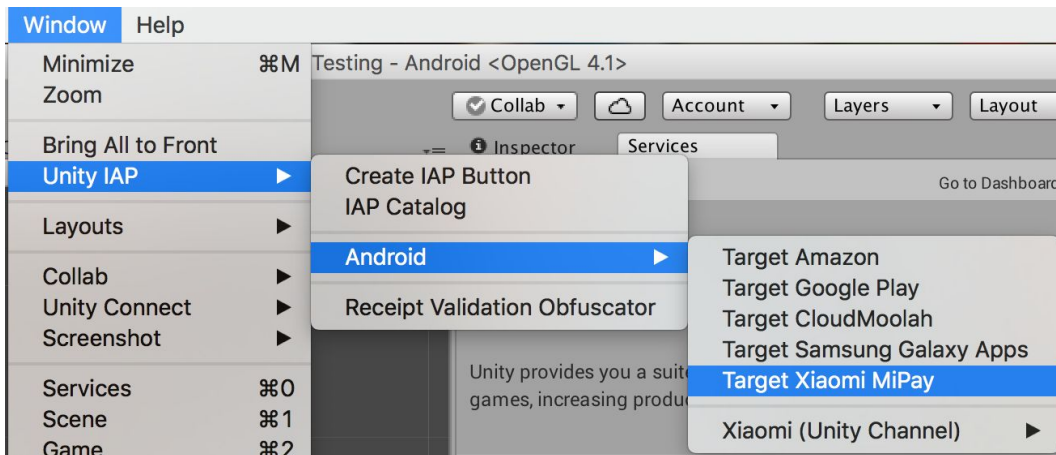
Install the “UnityChannel + Xiaomi UnityIAP beta plugin” in the game project.

Look at the sample script in /Assets/Plugins/UnityPurchasing/script/IAPDemo.cs. This is copied into the Unity game project when the UnityIAP plugin is installed.

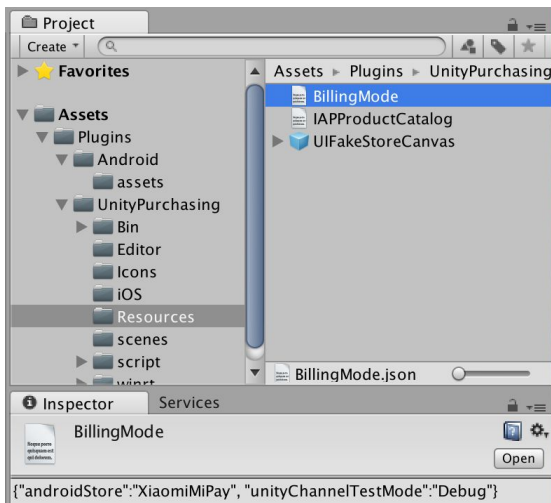
See [Getting Started with Unity IAP](#) to enable the “In-App Purchasing Service” for the project.



To build APK for Xiaomi MiPay store (not Samsung GALAXY, or GooglePlay) target the store with menu, “Window” -> “Unity IAP” -> “Android” -> “Target Xiaomi MiPay”. This enables Xiaomi + UnityChannel Java SDK. Use this to target different supported Android app stores, before creating an APK build.



This is saved in a file:



Below will show some code samples for how to use Unity IAP, all the details could be found in the IAPDemo.cs

2.3.1 SDK Initialization

```
private IUnityChannelExtensions m_UnityChannelExtensions;
```

```
public void OnInitialized (IStoreController controller, IExtensionProvider extensions)
{
    m_UnityChannelExtensions = extensions.GetExtension<IUnityChannelExtensions> ();
}
```

2.3.2 SDK Binding

```
AppInfo unityChannelAppInfo = new AppInfo();
// Xiaomi appId
unityChannelAppInfo.appId = "abc123";
// Xiaomi appKey
unityChannelAppInfo.appKey = "efg456";
// UnityChannel Client Id
unityChannelAppInfo.clientId = "hij789";
// UnityChannel Client Key
unityChannelAppInfo.clientKey = "klm012";

unityChannelLoginHandler = new UnityChannelLoginHandler();
unityChannelLoginHandler.initializeFailedAction = (string message) =>
{
    Debug.LogError("Failed to initialize and login to UnityChannel: " + message);
};
unityChannelLoginHandler.initializeSucceededAction = () =>
{
    //...
};
StoreService.Initialize(unityChannelAppInfo, unityChannelLoginHandler);
```

2.3.3 SDK Login

```
// The login succeeded callback will the userInfo as the parameter, in which:
// string userInfo.channel: the channel name, e.g. 'XIAOMI'
// string userInfo.userId: the channel name and the channel uid,
// string userInfo.userLoginToken: Unity user login token used for server side
// 'verifyLogin' api
unityChannelLoginHandler.loginSucceededAction = (UserInfo userInfo) =>
{
    m_IsLoggedIn = true;
    // userInfo.channel
```

```

// userInfo.userId
// userInfo.userLoginToken
};

unityChannelLoginHandler.loginFailedAction = (string message) =>
{
    m_IsLoggedIn = false;
    Debug.LogError("Failed logging into UnityChannel. " + message);
};

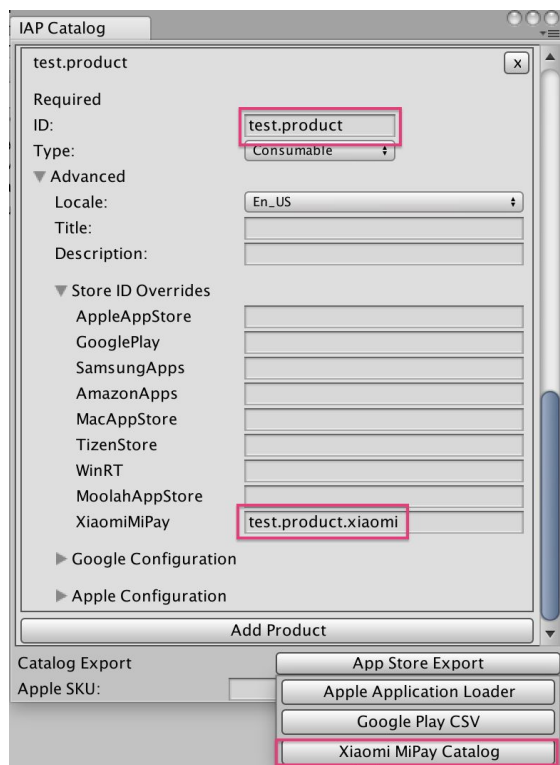
StoreService.Login(unityChannelLoginHandler);

```

2.3.4 Catalog Setting

On the menu, “Window” -> “Unity IAP” -> “IAP Catalog”. Set the product “ID”.

In the “Advanced” -> “Store ID Overrides”, set the product id in “XiaomiMiPay”. Note: Default “XiaomiMiPay” value is the same as for product “ID”.



Next, to use the IAP Catalog product list in the script of the game project, either:

- a) Read the catalog file. Remove all calls to “builder.AddProduct(…)” in your copy of IAPDemo.cs and replace that code with this code which will read the IAP Catalog:

```
var catalog = ProductCatalog.LoadDefaultCatalog();
foreach (var product in catalog.allProducts) {
    if (product.allStoreIDs.Count > 0) {
        var ids = new IDs();
        foreach (var storeID in product.allStoreIDs) {
            ids.Add(storeID.id, storeID.store);
        }
        builder.AddProduct(product.id, product.type, ids);
    } else {
        builder.AddProduct(product.id, product.type);
    }
}
```

Or:

- b) Manually input each product in the catalog. Add calls to “builder.AddProduct(…)” for each Product ID in IAP Catalog.

2.3.5 Purchase Validation

// For stand-alone game, please use the ValidateReceipt api to validate purchase locally,
// For online game, please get the orderQueryToken and validate purchase with server
side api,

// The callback of ValidateReceipt will take these parameters:

// bool result: whether purchase succeeded, e.g. ‘true’

// string signData: the order content which is in Json String format (details can be seen in
the table below)

// string signature: the RSA signature with the signData. Please use the Unity RSA public
key to validate the signData

```
public PurchaseProcessingResult ProcessPurchase(PurchaseEventArgs e)
{
    var unifiedReceipt = JsonUtility.FromJson<UnifiedReceipt>(e.purchasedProduct.receipt);
```



```

if (unifiedReceipt != null && !string.IsNullOrEmpty(unifiedReceipt.Payload))
{
    var purchaseReceipt =
        JsonUtility.FromJson<UnityChannelPurchaseReceipt>(unifiedReceipt.Payload);
    // get orderQueryToken
    var orderQueryToken = purchaseReceipt.orderQueryToken;
    // or do ValidateReceipt locally
    m_UnityChannelExtensions.ValidateReceipt(
        purchaseReceipt.gameOrderId
        (bool result, string signData, string signature) => {
            // validate the signData with signature and Unity RSA public key
        });
}
}

```

signData content:

Attribute Name	Required?	Description	Format	Sample
orderAttemptId	Yes	orderAttempt id assigned by Unity	String	274877943826
status	Yes	status of the orderAttempt	String	SUCCESS
productId	Yes	product id of the product associated with the orderAttempt	String	Product_1
quantity	Yes	quantity of the product	Integer	1
extension	No	extesion	Json String	{"abc" : "123"}
paidTime	Yes	orderAttempt paid time	yyyy-MM-dd hh:mm:ss, GMT+8 timezone	2017-03-08 06:43:20

cpOrderId	Yes	order id assigned by Game	String	66mea52wne
clientId	Yes	client id returned after Game onboarding to Unity Channel	String	P-wU0n9pbyQ1ulks4kHX_Q
country	Yes	currency	String of ISO 4217	CNY
currency	Yes	country	String of ISO 3166-2	CN
payFee	Yes	pay fee of this order (in channel unit)	String	1

