

Developer  
/ Documentation  
/ Unreal Engine ▾  
/ Unreal Engine 5.4 Documentation  
/ Programming and Scripting  
/ Online Subsystems and Services  
/ Online Services  
/ Online Services Interfaces  
/ Commerce Interface

# Commerce Interface

Purchase and redeem game content outside of gameplay.



! Learn to use this **Beta** feature, but use caution when shipping with it.

The **Online Services Commerce Interface** is Unreal Engine's foundation for any game service that provides players the ability to purchase or redeem game content outside of gameplay.

The Commerce Interface consists of two primary components:

- **Transactions:** the process of purchasing store items with platform currency.
  - Once the transaction is complete, the interface will grant the player the appropriate entitlements.
- **Entitlements:** content that a player is entitled to receive or use.
  - A player may receive or use an entitlement because they purchased an item or redeemed a game code.

## API Overview

The following table provides a high-level description of the functions included in the Commerce Interface.

Function	Description
<b>Offers</b>	
<code>QueryOffers</code>	Fetch the list of all available offers from the store and cache them in the interface. This includes any available downloadable content (DLC), bundles, items, and so on.
<code>QueryOffersById</code>	Fetch the list of available offers from a list of supplied IDs and cache them in the interface.
<code>GetOffers</code>	Retrieve offers from the interface cached by <code>QueryOffers</code> .
<code>GetOffersById</code>	Retrieve offers from a list of supplied IDs cached with the interface.
<b>Store</b>	
<code>ShowStoreUI</code>	Show the native store UI for the user to view store information or process transactions outside of the game client.
<b>Checkout</b>	
<code>Checkout</code>	Initiate the purchase process with one or more purchase offers retrieved with <code>GetOffers</code> or <code>GetOffersById</code> .
<b>Event Listening</b>	
<code>OnPurchaseCompleted</code>	Event that fires whenever a local user completes a transaction. This can be initiated either internally by <code>Checkout</code> or externally through the native store UI.

Function	Description
<b>Entitlements</b>	
<code>QueryTransactionEntitlements</code>	View the in-game entitlements corresponding to a successful <code>Checkout</code> call to provide these benefits to the player.
<code>QueryEntitlements</code>	Fetch the list of already acquired entitlements for the specific user from the store and cache them in the interface.
<code>GetEntitlements</code>	Retrieve entitlements from the interface cached by <code>QueryEntitlements</code> .
<code>RedeemEntitlement</code>	Mark an entitlement as "redeemed". The entitlement will then have the redeemed flag when later queried. This is useful for when there is no external game service managing entitlements.
<b>Verification</b>	
<code>RetrieveS2SToken</code>	Return a token that can be sent to game services backends to communicate with the platform and verify the ownership of the given entitlements.

## Process Flow

We now provide an example process flow using the Online Services Commerce Interface that regulates the flow of a user launching a game, entering the in-game store to make a purchase, and verifying that purchase for use.

## Launch Game

After a user launches the game and successfully authenticates with their desired online service, the game calls `QueryEntitlements`. The game compares the data cached by

`QueryEntitlements` against the registered entitlements in the user's save data to see which entitlements have been granted while the user was offline and appropriately applies them to the user. Simultaneously, the game uses the `OnPurchaseCompleted` event to listen for a message if the user completes a purchase in the future.

## Enter In-Game Store Interface

The user opens the menu for the store while in-game. The game shows a loading screen for the user opening the store while it calls `QueryOffers`. Once the query resolves, the game then calls `GetOffers` to get local copies of the data. The local copies are then passed on to the UI framework to render and display the game offers.

## Perform Transaction

Upon viewing the items on offer, the user decides to purchase a specific product, we will refer to this as `PRODUCT_A`. The user adds `PRODUCT_A` to their in-game cart (handled by the in-game UI) and confirms the transaction. After authenticating the user, the game calls `Checkout` with the ID of `PRODUCT_A`. This leads into the platform UI for final confirmation and payment handling.

Once `Checkout` successfully resolves and an `OnPurchaseCompleted` event fires, the game calls `QueryTransactionEntitlements` on the given transaction ID to obtain the in-game entitlement IDs granted to the user as part of the transaction, and applies them to the user's save game. If `PRODUCT_A` is not something that should be granted to the user's gameplay globally, the game then calls `RedeemEntitlement` to ensure that the entitlement for `PRODUCT_A` is not duplicated.

## Verify Transaction

After successfully purchasing and redeeming `PRODUCT_A`, the user decides to enter online play with their newly purchased `PRODUCT_A`. While authenticating the game server, the game notices that the local user is claiming a new entitlement and requests a verification token to ensure that the product is legitimate. The game calls `RetrieveS2SToken` on the IDs provided for verification and obtains a JSON Web Token (JWT), which the backend service then uses

to connect to the platform service and verify the ownership of the product. Once this returns successfully, the user is then permitted to enter online play with their newly purchased item.

## Converting Code from the Online Subsystem

The Online Services Commerce Interface is responsible for all code owned by both the **Store** (read-only code) and **Purchase** (read / write code) interfaces from the [Online Subsystem](#). The following table shows the correspondence between objects from the Online Systems Commerce Interface and their counterparts from the older Online Subsystem.

Online Services		Online Subsystem	
Interface	Object	Interface	Object
Commerce	Offer	<a href="#">Store</a>	Offer
Commerce	Entitlement	<a href="#">Purchase</a>	Entitlement

## More Information

### Header File

We welcome you to consult the `Commerce.h` header file directly for more information as needed. The Commerce Interface header file `Commerce.h` is located in the following directory:

```
Engine\Plugins\Online\OnlineServices\Source\OnlineServicesInterface\Public\Onlin
```

 Copy full snippet

For a step-by-step guide on how to obtain the UE source code, refer to our documentation on [Downloading Unreal Engine Source Code](#).

# Function Parameters and Return Types

Refer to the [Functions](#) section of the [Online Services Overview](#) page for an explanation of function parameters and return types, including how to pass parameters and processing the results when functions return.