

# Store Interface

Online subsystems for describing and filtering offers for in-game purchases.



Two interfaces power your game's ability to support user purchases: The **Store Interface**, which provides the ability to make offers to your users, and the **Purchase Interface**, which enables users to accept these offers. With the Store Interface, the game can retrieve offers from the online service, place them into categories, filter them, and manage them on an individual basis. To execute a purchase, use the [Purchase Interface](#).

An offer (class `FOnlineStoreOffer`) contains all necessary data about anything a user can purchase through the game. This includes information like the description, the price, and the release and expiration dates. Categories (class `FOnlineStoreCategory`) hold much simpler data, containing only a name and a list of subcategories. The online services themselves handle setup and management of offers, while the Store Interface takes care of retrieval and filtration.

## Offers and Categories

The Store Interface retrieves offer and category data from the online service's servers. Because this operation involves contacting a remote machine, it is broken into two phases: Retrieval and caching of data, which is asynchronous, and accessing that locally-cached

data. The Store Interface itself will automatically update the cache whenever a request completes, keeping only the results of the most recent query.

## Retrieving Offers and Categories

The first step in retrieving offer information is to retrieve the list of categories that the store contains. The `QueryCategories` function handles this, and calls a provided delegate of type `FOnQueryOnlineStoreCategoriesComplete` when it finishes. If successful, the Store Interface will have a cache containing all available category IDs. At this point, the developer can begin querying for offers by making a filter (of type `FOnlineStoreFilter`) that includes certain categories, and excludes others. With a filter, `QueryOffersByFilter` can retrieve offer IDs based on their category membership. Once the Store Interface has retrieved and cached the offer IDs, developers can make queries for information about specific offers with the `QueryOffersById` function. Both `QueryOffersByFilter` and `QueryOffersById` will use a provided delegate of type `FOnQueryOnlineStoreOffersComplete` to indicate success or failure, and the Store Interface's cache of offers will update following either operation completing successfully.

## Examining Offer Data

The "query" functions retrieve data about categories and offers into the Store Interface's cache. To access this data, call `GetCategories` for the category cache, or `GetOffers` for the offer cache. If you have a known offer ID, the `GetOffer` function will return information about the corresponding (cached) offer.



These functions all act on the local cache, so they do not have a callback. However, they will only be useful if the cache has been populated. These functions will not update the cache, so any live changes to categories or offers on the online service's store will not be reflected.