

Microsoft.StoreServices Client Sample

*This sample is compatible with the Microsoft Game Development Kit (March 2022)*

# Description

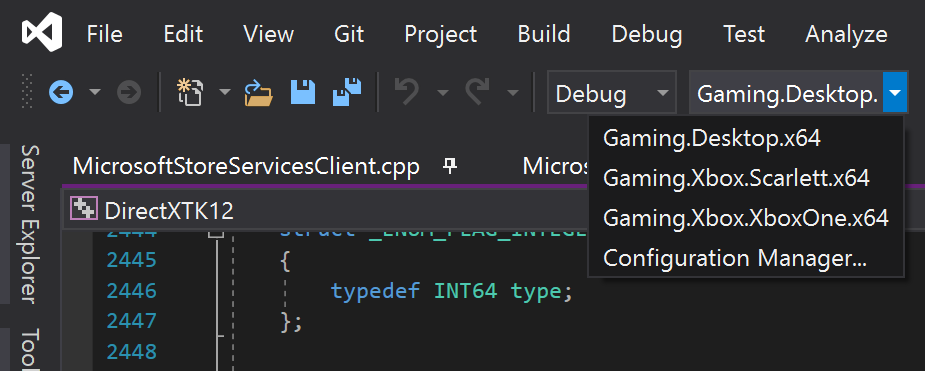
This sample demonstrates the client-based operations used with the [Microsoft.StoreServices Sample](https://github.com/microsoft/Microsoft-Store-Services-Sample) for service-to-service authorization and product management with the Microsoft Store Services. Specifically how to obtain and use the User Store Ids as outlined in [Requesting a User Store ID for service-to-service authentication (microsoft.com)](https://developer.microsoft.com/en-us/games/xbox/docs/gdk/xstore-requesting-a-userstoreid)

Graphical user interface

Description automatically generated

# Building the sample

This sample supports Xbox One, Scarlett as well as Desktop. Select the config in the dropdown to build.



*For more information, see* Running samples*, in the GDK documentation.*

# Running the sample

The sample is designed to work with the [Microsoft.StoreServices Sample](https://github.com/microsoft/Microsoft-Store-Services-Sample) and exercises the required flow and API’s on the client side required to allow your service to authenticate with the Microsoft Store Services. When running the sample in XDKS.1 it is pre-configured to call and interact with a version of the Microsoft.StoreServices Sample that ATG maintains for use with the sample products in XDKS.1.

The client sample will interact with the Service Sample to request specific actions and display the result of those actions done by the Service sample. For example, viewing the items the account owns, fulfilling consumable products, managing subscriptions, and tracking consumable fulfillments to detect refunds.

A key characteristic of using the XStore API’s with this client is that they require a valid license in order to function. Refer to the GDK documentation section titled “Setting up your product to test the XStore API” for full details.

A license is verified with a call to the licensing service when the game is launched. Without this available, XStore APIs (XStoreGetUserCollectionsIdAsync and XStoreGetUserPurchaseIdAsync) return 0x803f6107 (IAP\_E\_UNEXPECTED) indicating that a valid license was not found.

To obtain a valid license for the sample for your test account, ensure you are in sandbox **XDKS.1** (any developer account can use this sandbox), go the store page for the sample product and select **“Get”**. It is not necessary to wait for the download to complete.

To go directly to the store page for this sample, use the following shortcut commands:

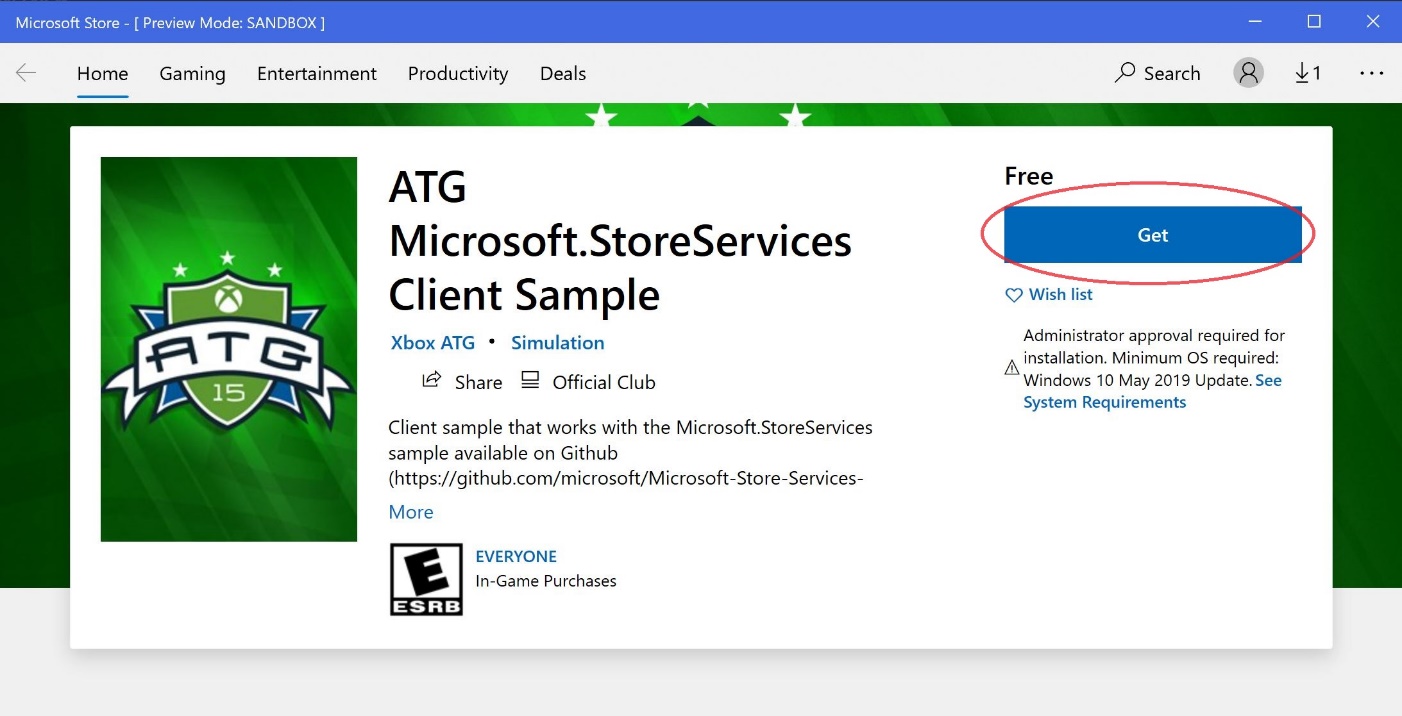
**PC:**

Run box (Win+R):

ms-windows-store://pdp/?productid=9MXL21XPWWWK

or

msxbox://game/?productId=9MXL21XPWWWK



**Xbox:**

In the Visual Studio Gaming Command Prompt:   
xbapp launch ms-windows-store://pdp/?productid=9MXL21XPWWWK

Graphical user interface

Description automatically generated

The sample as installed from the store will be properly licensed and function properly but may represent an older version of the sample.

The sample should now be able to run from Visual Studio with (F5) in the XDKS.1 sandbox. If you plan to run the sample in your own sandbox and as your own title for testing, you will need to make specific configuration changes to the MicrosoftGameConfig.mgc file as outlined in the section below.

The sample does not provide a way to purchase the sample products unless re-purchasing a consumable or subscription that the test account already had previously. You can use the In-Game Store sample to purchase various products or use the following shortcuts::

**PC:**

Run box (Win+R):

* **Store Managed Consumable Products:**ms-windows-store://pdp/?productid=9PFL4RQTB1P6  
  ms-windows-store://pdp/?productid=9NCX1H100M18
* **Durables without a package:**ms-windows-store://pdp/?productid=9N30KZZF4BR9  
  ms-windows-store://pdp/?productid=9P23V43P0XZZ  
  ms-windows-store://pdp/?productid=9PLRFWZWWF91
* **Game:**ms-windows-store://pdp/?productid=9NTL0QDWZ4FS
* **Subscription:**ms-windows-store://pdp/?productid=9MZ0MGGFPLTP

**Xbox:**

In the Visual Studio Gaming Command Prompt:

* **Store Managed Consumable Product:**xbapp launch ms-windows-store://pdp/?productid=9PFL4RQTB1P6  
  xbapp launch ms-windows-store://pdp/?productid=9NCX1H100M18
* **Durables without a package:**xbapp launch ms-windows-store://pdp/?productid=9N30KZZF4BR9  
  xbapp launch ms-windows-store://pdp/?productid=9P23V43P0XZZ  
  xbapp launch ms-windows-store://pdp/?productid=9PLRFWZWWF91
* **Game:**xbapp launch ms-windows-store://pdp/?productid=9NTL0QDWZ4FS
* **Subscription:**xbapp launch ms-windows-store://pdp/?productid=9MZ0MGGFPLTP

# Running the sample as your title

You can redirect the sample to use your title configuration in order to test and troubleshoot Note that this takes the place of any installed build, so be aware as installing the sample as your title may incur subsequent reinstallation cost.

1. Log into your sandbox and use a test account that is provisioned for sandbox
2. Ensure the test account owns the title so it has a digital license to run the title
3. From your title’s MicrosoftGameConfig.mgc. copy over
   1. Identity node; version doesn’t matter
   2. Title ID
   3. Store ID
   4. **Xbox only:** ContentIdOverride and EKBIDOverride (see below)
4. Rebuild and deploy
5. Launch (in your developer sandbox)

It is recommended you do a clean rebuild and uninstall all previous installations of the sample that points to any other title.

If you encounter issues on Xbox, do an xbapp list /d on the deployment and ensure that all values match the expected values of your title’s, aside from the names and version numbers you hadn’t changed. For both platforms, ensure the PFN matches in terms of the app identity as well as the suffix which is a function of your publisher.

**Xbox only:**

A **locally deployed build** (i.e. push or run from PC) will not be licensable by default, but if the MicrosoftGameConfig.mgc contains development only override values for content ID and EKBID, it will be able to license properly and allow XStore API to work. Note this section in the MicrosoftGameConfig.mgc in this sample:

<DevelopmentOnly>

<ContentIdOverride>2797FA46-A93B-494C-AD80-B67C9FCA939F</ContentIdOverride>

<EKBIDOverride>00000000-0000-0000-0000-000000000001</EKBIDOverride>

</DevelopmentOnly>

Content ID must match that assigned to the package submitted the sandbox in Partner Center.

EKBID can be anything other than all zeroes or the default 33EC8436-5A0E-4F0D-B1CE-3F29C3955039.

Once this is in place, and in combination with an account licensed to the product, the sample will run in licensed state.

The best way to obtain the content ID, the proper EKBID (not required), and PFN is to install the ingested and published package from sandbox and then running xbapp list /d

Registered Applications by Package Full Name:

41336MicrosoftATG.InGameStoreXS\_1.0.0.0\_neutral\_\_dspnxghe87tn0

Install

Drive: Retail

Size: 0.28 GB.

ContentId: {2797FA46-A93B-494C-AD80-B67C9FCA939F}

ProductId: {4C544E39-5130-3044-C057-5A3446536A00}

EKBID: {37E80840-6BEE-46F8-8EDB-92F877056087}

DisplayName: ATG In-Game Store Sample

41336MicrosoftATG.InGameStoreXS\_dspnxghe87tn0!Game

These values can also be seen onscreen by selecting Menu on the installed title’s tile in My Games and looking at File Info.

In the case of EKBID, this is visible upon package registration, i.e. Ready to Launch, so if your title’s package is large, this can be cancelled at this time once you have the EKBID and intend to deploy or sideload your development build instead.

# Implementation notes

The Microsoft Account (MSA) the StoreContext and therefore the UserStoreIDs are tied to depends on if the app is running on Windows or on an Xbox Console.

* **Windows PC:** The MSA that is signed into the Windows Store App, not necessarily the MSA signed into Xbox Live.
* **Xbox:** The MSA that is signed into Xbox Live and is actively playing the game.

For more on this see the documentation [Handling mismatched store account scenarios on PC (microsoft.com)](https://developer.microsoft.com/en-us/games/xbox/docs/gdk/xstore-handling-mismatched-store-accounts)

## Paging

If the results from the Collections or Subscriptions pages include more than 4 results, the **Next Page** button will be enabled to allow you to go through all the items that were returned. Your current page # and total pages are also shown.

## Refresh UserStoreIds

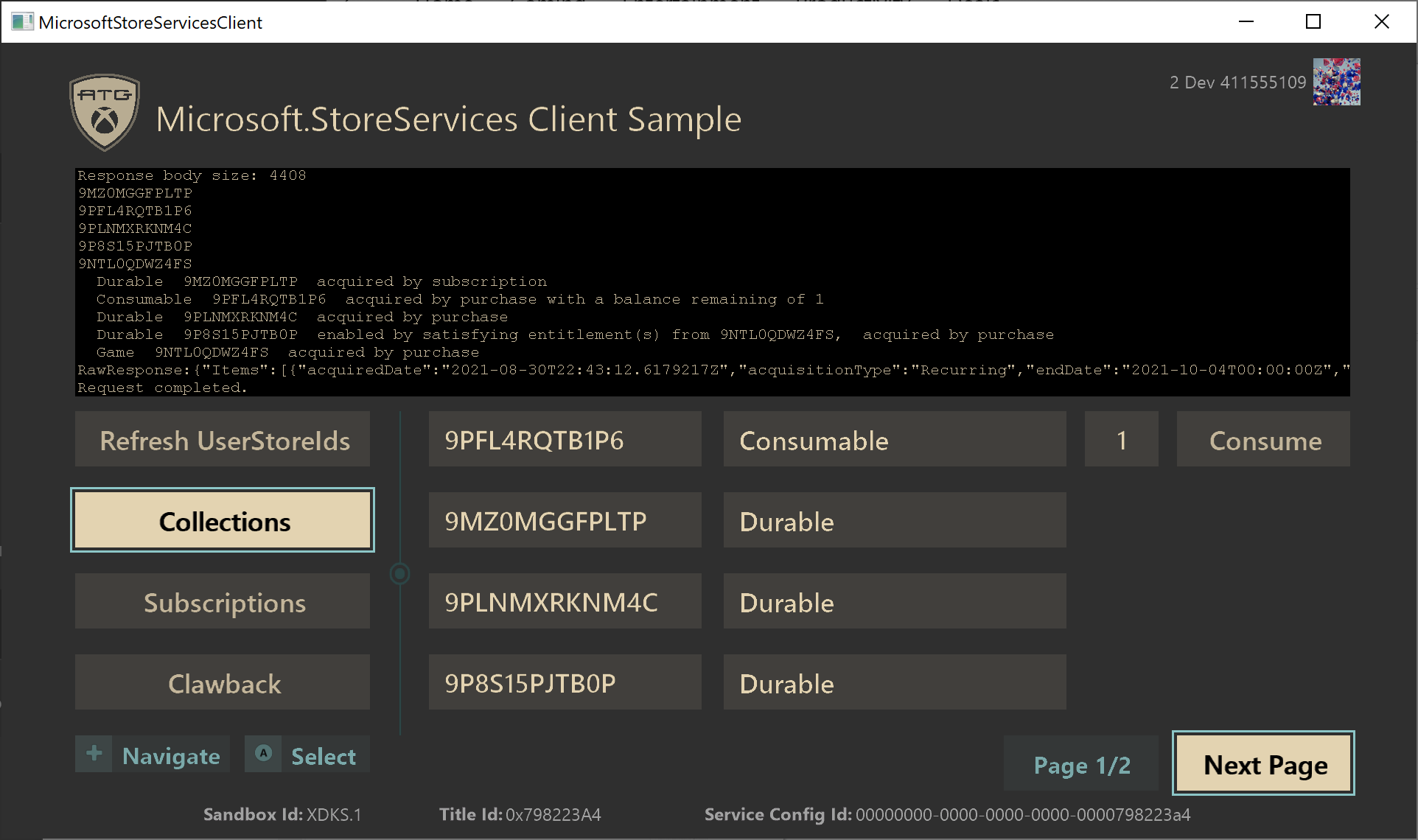
This button will begin the process of retrieving the AAD Access Tokens from the Service Sample and then use those to generate the UserStoreIds (UserCollectionsId and UserPurchaseId) that will need to be handed to the Service Sample to preform the service-to-service auth for the Microsoft Store Services

For more information see [Requesting a User Store ID for service-to-service authentication (microsoft.com)](https://developer.microsoft.com/en-us/games/xbox/docs/gdk/xstore-requesting-a-userstoreid)

## Collections

On the Collections page, a service-to-service query is made by the Service Sample to check for the user’s Collections data. This includes the items the user has purchased such as Games, Durables, and Consumables. If the user has a consumable product in their query results you can do the following actions within the client sample:

* **Consume the quantity** - If the quantity is greater than 0 - Fulfill the item from the user’s account and add the value to the user’s balance on our own Service’s database for tracking consumable purchases for the user and possible refunds issue to the user for these fulfilled items. This consume is also added to the consumable and Clawback tracking built-into the Service Sample.
* **Purchase more of the consumable -** If the quantity is 0 - Purchase more of the consumable directly from within the app to continue testing and simulate multiple purchases of the same consumable.



More information see the sections under [Manage products from your services (contents) (microsoft.com)](https://developer.microsoft.com/en-us/games/xbox/docs/gdk/service-to-service-nav)

## Subscriptions

On the Subscriptions page, a service-to-service query is made by the Service Sample to check for the user’s subscriptions. If the user has a subscription product in their query results you can do the following actions within the client sample:

* **Postpone –** Active subscription with auto-renew enabled – This turns off the auto-renew setting of the subscription. This allows the user to finish their remaining time of their subscription, but it will become invalid after the end date.
* **Cancel –** Active subscription with auto-renew disabled – This cancels the user’s subscription immediately and changes it from Active to Inactive regardless of the time left on their currently paid subscription period.
* **Renew –** Inactive subscription – Initiate the purchase flow to re-subscribe to the subscription product.

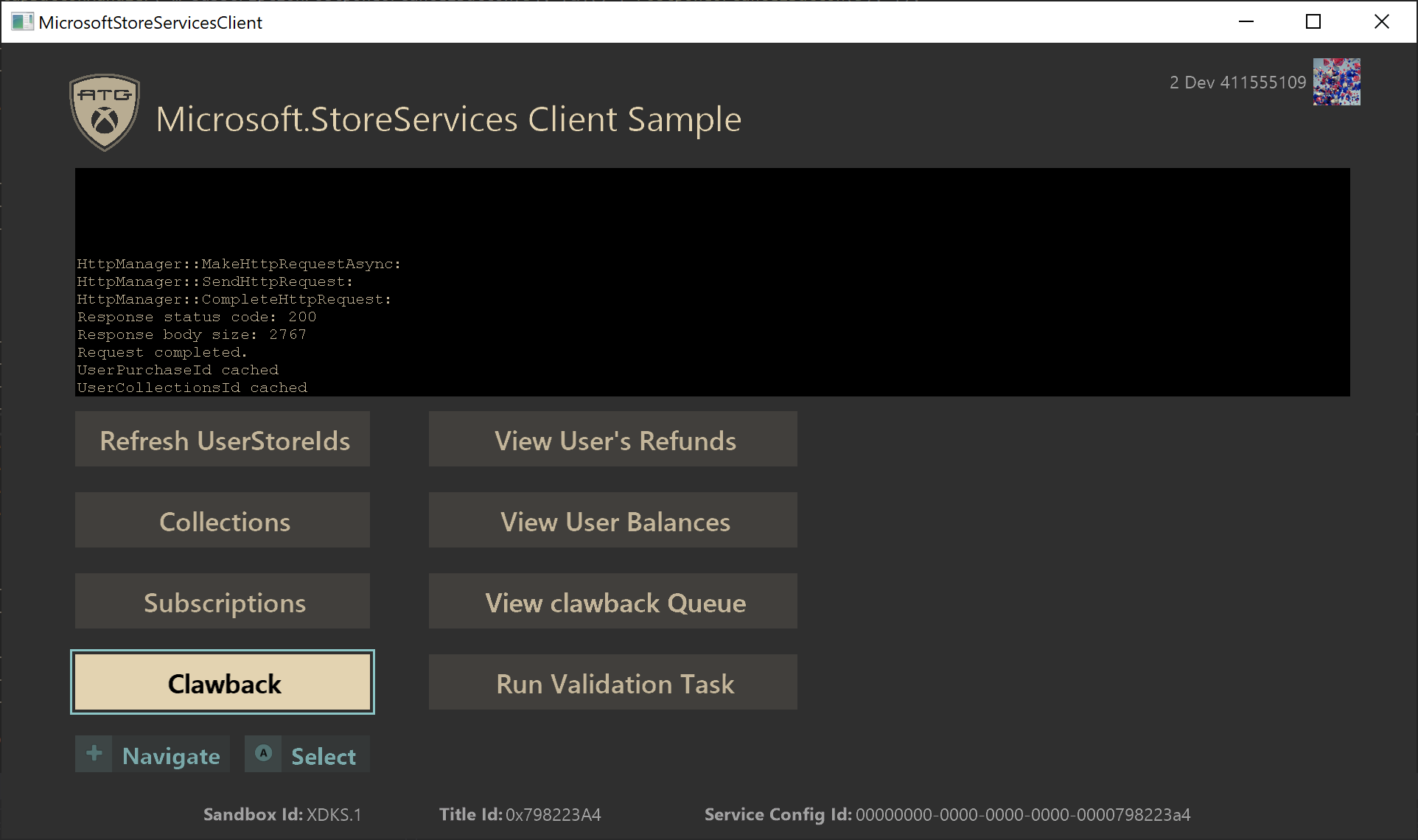
Graphical user interface, text

Description automatically generated

## Clawback

On the Clawback page, you have the following buttons to request data or actions from the Service Sample:

* **View User’s Refunds** – Service Sample will preform a call to check if the current user has any refunded items.
* **View User Balances** – View the currently tracked balance of fulfilled consumables on the service for all users.
* **View Clawback Queue** – View the currently tracked consume transactions to look for possible refunds with the Clawback service.
* **Run Validation Task** – Service Sample will preform the Clawback Reconciliation to look for refunds of all users and tracked consumable transactions.



For more information on see [Managing consumable products and refunds from your service (microsoft.com)](https://developer.microsoft.com/en-us/games/xbox/docs/gdk/xstore-managing-consumables-and-refunds)

# Known issues

No currently known issues.

# Privacy statement

When compiling and running a sample, the file name of the sample executable will be sent to Microsoft to help track sample usage. To opt-out of this data collection, you can remove the block of code in Main.cpp labeled “Sample Usage Telemetry”.

For more information about Microsoft’s privacy policies in general, see the [Microsoft Privacy Statement](https://privacy.microsoft.com/en-us/privacystatement/).

# Update history

**Update:** June 2022

**Initial Release:** August 2021