

Microsoft Store Services Sample

# Description

This sample focuses on how your custom game service can authenticate and call into the Microsoft Store Services including Collections and Purchase endpoints. Specifically, this sample introduces and demonstrates how to use the Azure Active Directory (AAD) auth flow to generate Access Tokens and User Store Ids to call these services.

Highlighted are the added features that are not available through calling these services using XSTS tokens:

* Clawback – Ability to detect refunds issued to a user that have already been fulfilled within your consumable tracking system.
* Recurrence – Ability to view and manage user subscriptions.

There are two projects that make up the sample:

* **Microsoft.StoreServices** – Provides a namespace of API’s to generate the authentication flow and call the individual services. This is publicly available on GitHub for the latest version here:  
  <https://github.com/microsoft/Microsoft-Store-Services>
* **MicrosoftStoreServicesSample** – Sample service that mimics a game service that uses Microsoft.Store.Services but includes additional code to manage, cache, and track user commerce information.

The MicrosoftStoreServicesSample portion also provides an example implementation for tracking and managing consumable transactions as well as reconciling those transactions with any refunds issued to the user to revoke the items granted from those now refunded items.

# Building the Sample

See the included Configuration and Documentation Guide for step-by-step instructions to build and deploy the sample.

# Using the sample

See the included Configuration and Documentation Guide for instructions on how to configure and use the sample as a running service.

# Implementation notes

This sample has the following key features and design:

* Microsoft.StoreServices namespace that provides classes and controllers to authenticate with the Microsoft Store Services as outlined in [the following documentation](https://docs.microsoft.com/en-us/windows/uwp/monetize/view-and-grant-products-from-a-service). The most up to date version of this namespace is available on GitHub at <https://github.com/microsoft/Microsoft-Store-Services> .
* Classes and functions to call and use Collections, Clawback, and Recurrence
* Example code to manage and track consumable fulfillment within your game service
* Example code to reconcile fulfilled consumables against Clawback data indicating what has been refunded.

# Known issues.

* The sample does not currently monitor or handle throttled calls to the Microsoft Store Services. When calling a service too frequently calls will return with a 429 HTTP status and information on when the next request can be made.
* This is the initial release and there may be bugs or best-practice improvements that will be found and updated in future releases. If you find anything not working or that can be improved, please provide us feedback on the GitHub sites:
  + [microsoft/Microsoft-Store-Services (github.com)](https://github.com/microsoft/Microsoft-Store-Services)
  + [microsoft/Microsoft-Store-Services-Sample (github.com)](https://github.com/microsoft/Microsoft-Store-Services-Sample)
* This is a sample, and although it can be used as an example and some code used to build a running service, know that the service sample is not hardened or validated to run as a large, distributed service. You should always do your own review and rely on your own security and performance reviews that would meet the need of your services.

# Update history

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| March 2021 | 1.0 | * Initial release |
| April 2021 | 1.01 | * Fixed a few logic issues with Clawback. * Updated sample service code to use the architecture of the public GitHub repo for Microsoft.StoreServices |

# Privacy Statement

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