

PlayFab Characters

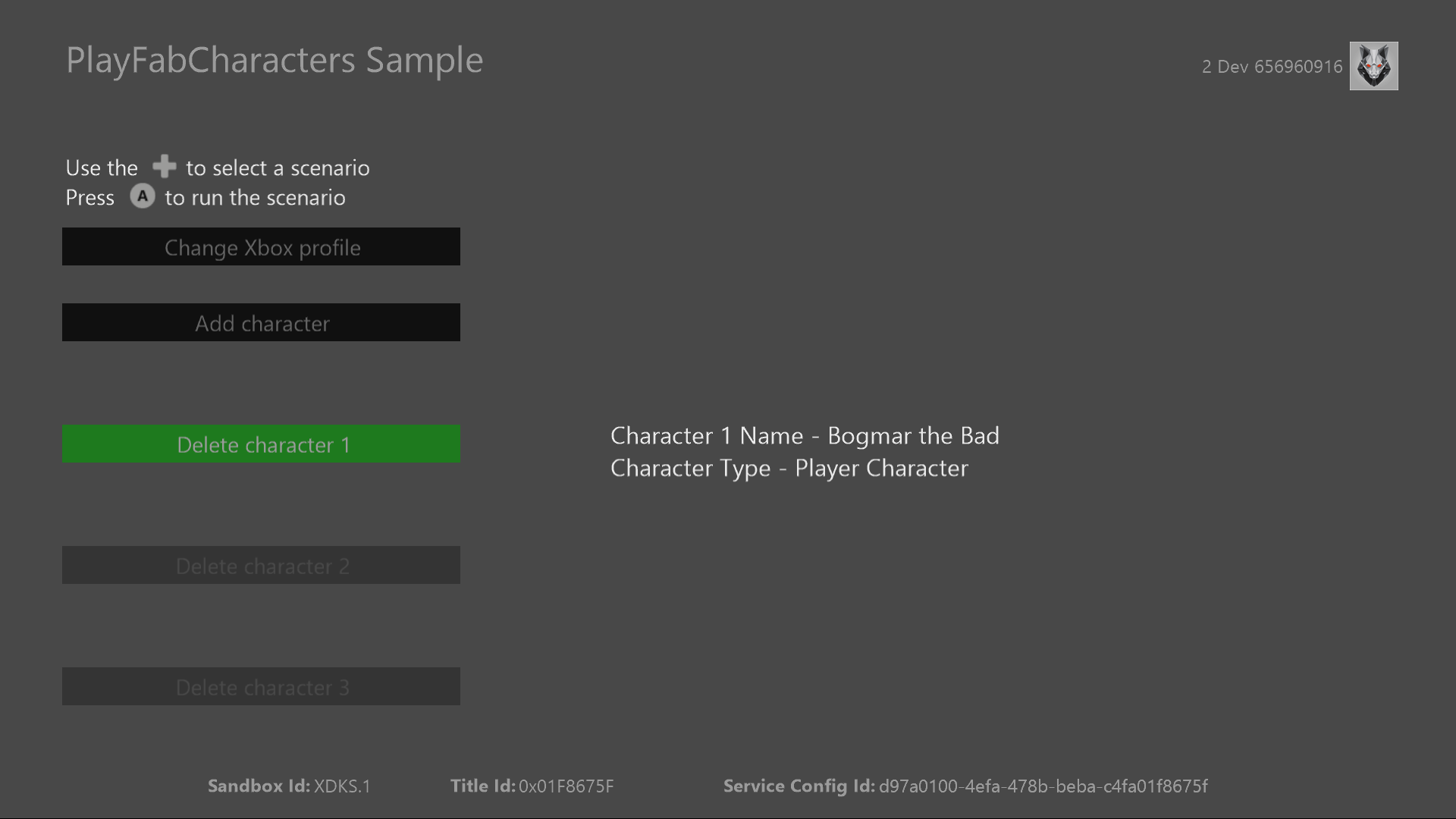
*\* This sample is compatible with the April 2018 XDK*

# Description

This sample demonstrates how to grant characters to, and delete them from, a player’s account.

# Using the sample

This sample is configured to work in the XDKS.1 sandbox.



|  |  |
| --- | --- |
| Action | Gamepad |
| Select operation | D-pad |
| Execute action | A button |
| Exit | View button |

# Implementation notes

In this version of the XDK, the sample also requires your Xbox One console to use the XDKS.1 **SandboxID**. To switch to this **SandboxID**, in the Xbox One XDK command prompt, use:

xbconfig sandboxid=XDKS.1

The PlayFab title ID must be set in the sample initialization code. This sample uses the PlayFab title ID “DC0” which is assigned to the **PlayFabSettings::titleId** static field.

# PlayFab Characters

Characters in PlayFab are an extension of a player’s own profile. Each character has a unique inventory, and their own stats and custom data. Since each character may be associated with sensitive information, such as paid items in its inventory, granting and deleting them requires either a transaction from client code or execution from a secure server.

This sample utilizes simple PlayFab cloud scripts to query a player’s characters, and grant and delete them when requested. The cloud scripts used to grant and delete characters are as follows:

handlers.grantCharacter = function (args, context) {

var request = {

PlayFabId: currentPlayerId, CharacterName: args.Name, CharacterType: "Player Character"

};

server.GrantCharacterToUser(request);

};

handlers.deleteCharacter = function (args, context) {

var request = {

PlayFabId: currentPlayerId, CharacterId: args.CharId, SaveCharacterInventory: false

};

server.DeleteCharacterFromUser(request);

};

Granting a character to a player requires a character name, which is similar to a player’s display name, and a character type. The player decides the name of the character to create, then the name is passed to the script in the **Name** argument to be used in the **GrantCharacterToUser** call.

**Note** Unique names are not enforced by the service. If you want uniquely named characters that must be enforced in title code.

Once a character is created it is give a unique identifier similar to a user’s PlayFab ID. This ID is required for **DeleteCharacterFromUser**. This sample allows the player to pick the character they want to delete, then the corresponding character ID is passed to cloud script in the **CharId** argument.

# Update history

**Initial Release:** June 2018

# 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/).