

PlayFabのキャラクター

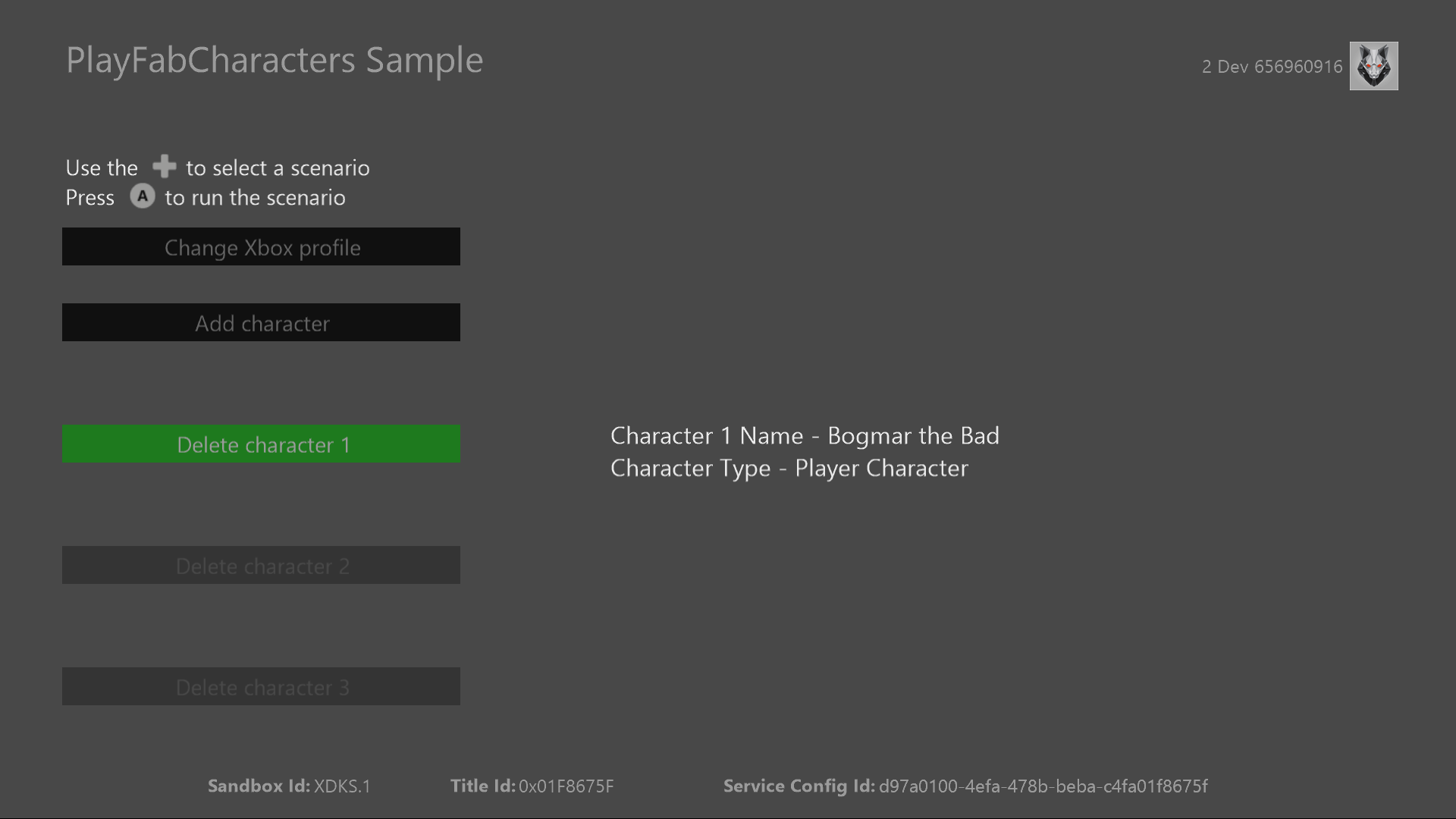
*\* このサンプルは、2018 年 4 月の XDK と互換性があります*

# 概要

このサンプルでは、プレイヤーのアカウントにキャラクターを付与したり、削除したりする方法を示しています。

# サンプルの使用

このサンプルは、XDKS.1 サンドボックスで機能するように構成されています。



|  |  |
| --- | --- |
| 作用 | ゲームパッド |
| 操作を選択 | D-Pad |
| アクションを実行 | Aボタン |
| 終了 | 表示ボタン |

# 実装上の注意

このバージョンの XDK では、このサンプルでは Xbox One 本体に XDKS.1 **SandboxID**を使用する必要があります。この **SandboxID**に切り替えるには Xbox One XDK コマンドプロンプトで、次のように入力します。

xbconfig sandboxid=XDKS.1

PlayFab のタイトル ID は、サンプル初期化コードで設定する必要があります。このサンプルでは、 **PlayFabSettings::titleId** 静的フィールドを割り当てた、PlayFabのタイトルID「DC0」を使用しています。

# PlayFabのキャラクター

PlayFabのキャラクターはプレイヤー自身のプロフィールの拡張です。各キャラクターはユニークなインベントリ、そして独自のステータスとカスタムデータを持っています。各文字はそのインベントリ内の有料アイテムなどの機密情報に関連付けられている可能性があるため、それらの許可および削除には、クライアントコードからのトランザクションまたは安全なサーバーからの実行が必要です。

このサンプルは単純なPlayFabクラウドスクリプトを利用してプレイヤーのキャラクターを照会し、要求されたらそれらを許可したり削除したりします。文字の付与と削除に使用されるクラウドスクリプトは次のとおりです。

handlers.grantCharacter = function (args, context) {

var request = {

PlayFabId: currentPlayerId, CharacterName: args.Name, CharacterType: 「プレイヤーキャラクター」

};

server.GrantCharacterToUser(request);

};

handlers.deleteCharacter = function (args, context) {

var request = {

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

};

server.DeleteCharacterFromUser(request);

};

プレイヤーにキャラクターを付与するには、プレイヤーの表示名に似たキャラクター名とキャラクタータイプが必要です。プレイヤーは作成するキャラクターの名前を決め、それから名前が **GrantCharacterToUser** 呼び出しにおいて**名** で使用される引数のスクリプトに渡されます 。

**注意** 一意の名前はサービスによって強制されません。あなたが呼び出しコードで強制されなければならない一意の名前の文字が必要な場合。

キャラクターが作成されると、ユーザーの PlayFab ID に似た一意の識別子が与えられます。このIDは **DeleteCharacterFromUser**で必要とされます。このサンプルはプレイヤーが削除したいキャラクターを選ぶことを可能にし、対応するキャラクターIDは **CharId** 引数にあるクラウドスクリプトに渡されます。

# 更新履歴

**初回リリース：**2018 年 6 月

# プライバシーステートメント

サンプルをコンパイルして実行すると、サンプルの実行可能ファイルの名前がMicrosoftに送信され、サンプルの使用状況の追跡に役立ちます。このデータ収集を無効にするには、Main.cppの「Sample Usage Telemetry」というラベルの付いたコードブロックを削除します。

マイクロソフトのプライバシーポリシー全般に関する詳細については、 [Microsoftのプライバシーステートメント](https://privacy.microsoft.com/ja-jp/privacystatement/)をご参照ください。