

PlayFab 角色

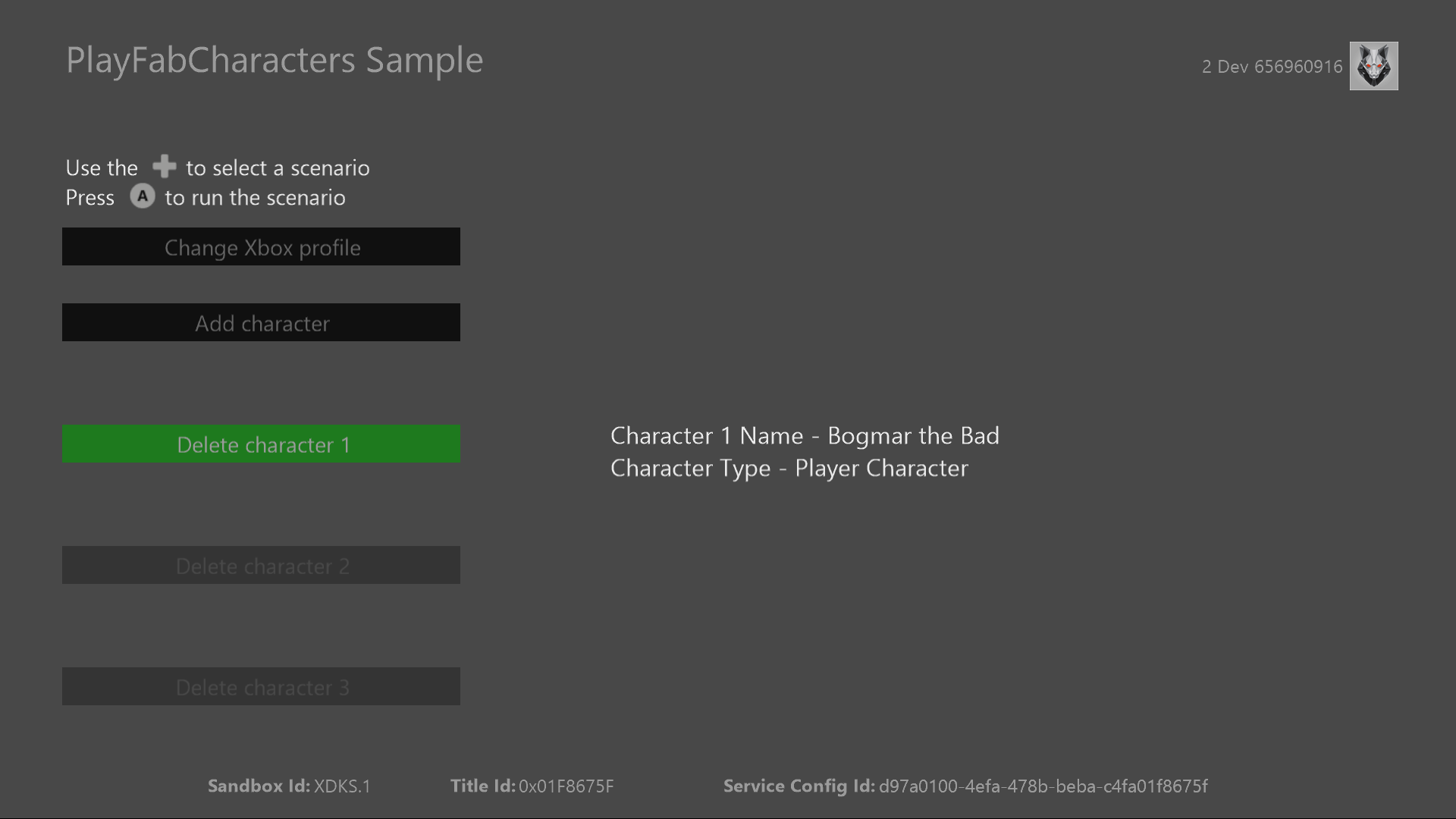
*\*此样本与2018年4月的XDK兼容*

# 描述

此样本演示如何向玩家的帐户授予角色，以及从中删除角色。

# 使用样本

此示例配置为在XDKS.1沙箱中工作。



|  |  |
| --- | --- |
| 操作 | 游戏手柄 |
| 选择操作 | D-Pad |
| 执行操作 | A 键 |
| 退出 | 查看按键 |

# 实施说明

在此版本的XDK中，该示例还要求您的Xbox One控制台使用XDKS.1 **SandboxID**。要切换到这个 **SandboxID**，在Xbox One XDK命令提示符下，使用：

xbconfig sandboxid = XDKS.1

必须在样本初始化代码中设置PlayFab标题ID。此示例使用分配给的PlayFab标题ID“DC0” **PlayFabSettings::titleId** 静态场。

# PlayFab 角色

PlayFab 中的角色是玩家自己的个人资料的扩展。每个角色都有一个独有的库存，及其自己的统计数据和自定义数据。由于每个角色可能与敏感信息相关联，比如，其库存中的付费项目，因此，授予和删除它们需要来自客户端代码的事务或来自安全服务器的执行。

此样本使用简单的 PlayFab 云脚本查询玩家的角色，并经请求授予和删除它们。用于授予和删除角色的云脚本如下：

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);

};

向玩家授予角色需要一个角色名称（类似于玩家的显示名称）和一个角色类型。玩家决定要创建的角色的名称，然后，将名称传递给 **Name** 变量中的脚本，该变量将用于 **GrantCharacterToUser** 调用。

**注意：**该服务不强制使用唯一名称。如果您需要唯一名称的角色，则必须在游戏代码中强制执行。

创建角色后，就会为其提供一个类似于用户 PlayFab ID 的唯一标识符。此 ID 是 **DeleteCharacterFromUser** 必需的。此样本允许玩家选择其想要删除的角色，然后，将相应的角色 ID 传递给 **CharId** 变量中的云脚本。

# 更新历史记录

**初始发行：**2018 年 6 月

# 隐私声明

编译和运行示例时，示例可执行文件的文件名将发送给Microsoft以帮助跟踪示例使用情况。要选择退出此数据收集，您可以删除Main.cpp中标记为“Sample Usage Telemetry”的代码块。

有关 Microsoft 隐私政策的更多信息，请参阅 [Microsoft 隐私声明](https://privacy.microsoft.com/zh-cn/privacystatement/)。