

简明型 PlayFab Auth

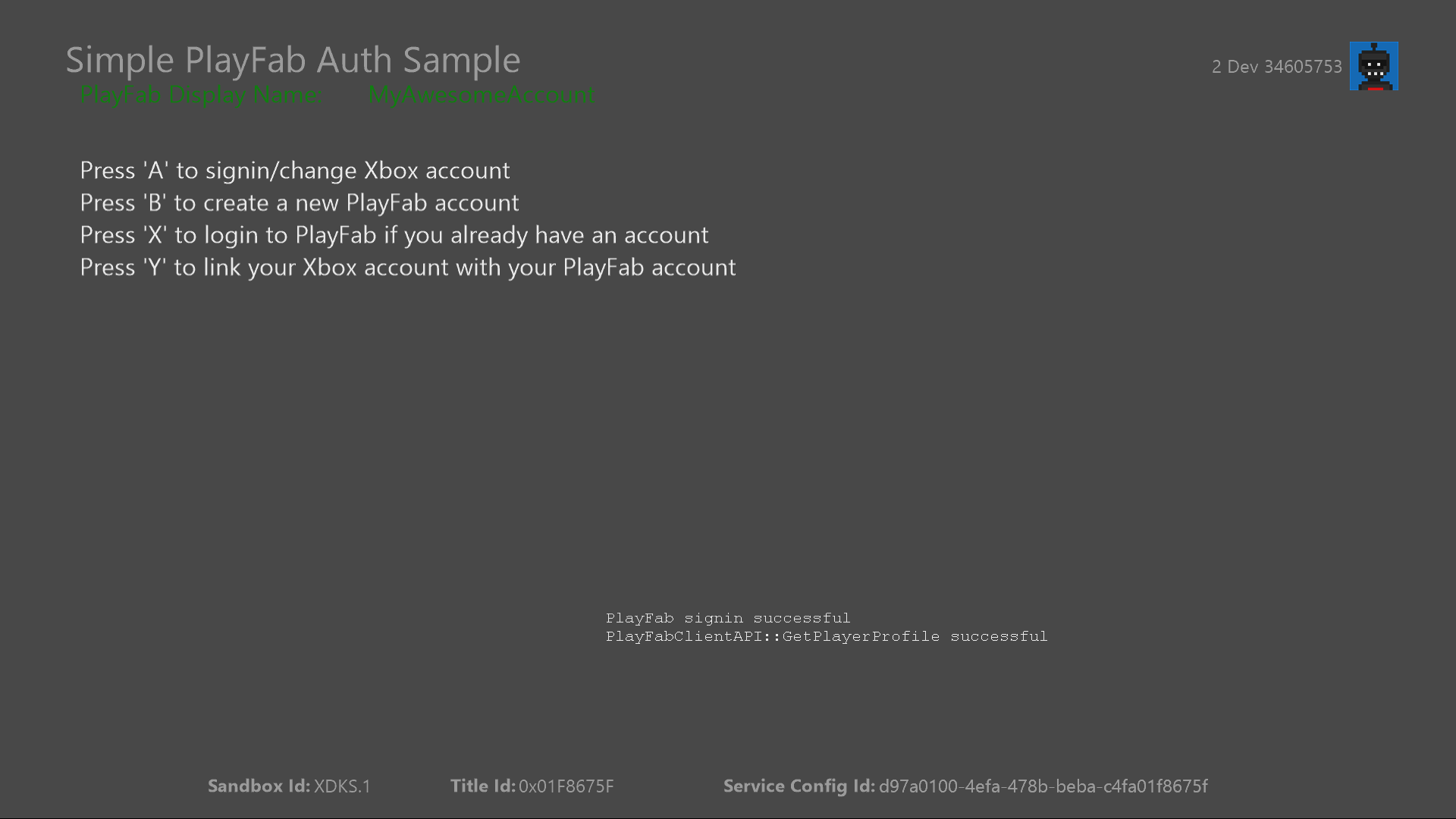
*\*此样本与2018年4月的XDK兼容*

# 描述

此示例演示了如何使用 PlayFab 注册 Xbox 用户，以便使用 PlayFab 服务。

# 使用样本

此示例配置为在XDKS.1沙箱中工作。



|  |  |
| --- | --- |
| 操作 | 游戏手柄 |
| 打开帐户选取器，选择 Xbox 配置文件 | A 键 |
| 注册新的 PlayFab 帐户 | B 键 |
| PlayFab 无缝登录 | X 键 |
| 将 Xbox 配置文件与现有 PlayFab 帐户关联 | Y 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。

# 使用 PlayFab 登录

Windows::Xbox::System::User::GetTokenAndSignatureAsync 🡪 PlayFab::PlayFabClientAPI::LoginWithXbox

通过 PlayFab 注册玩家最简单的方法是使用 Windows Xbox C++ PlayFab SDK 中的 **LoginWithXbox** API。只要登录 Xbox 配置文件，此方法就可用于以编程方式登录 PlayFab 帐户，而无需用户输入。如同所有 PlayFab API，LoginWithXbox 采用“请求”对象和回调（将在成功完成 HTTP 请求后执行）。它可以选择性地采用第二次回调来处理失败，并针对将传递到回调的任意数据采用 void 指针。此 API 需要一个管理登录选项的 **LoginWithXboxRequest** 对象。

|  |  |
| --- | --- |
| 字段 | 描述 |
| XboxToken | 使用端点 <https://playfabapi.com> 通过 **GetTokenAndSignatureAsync** 方法生成的令牌。 |
| CreateAccount | 如果设置为“true”，则 **LoginWithXbox** 将为 Xbox 配置文件创建一个新帐户（如果尚不存在）。  如果设置为“false”，则只有在 Xbox 配置文件已与 PlayFab 帐户关联时方可成功登录。否则，登录尝试将失败，并将执行错误回调。  此示例不强制创建新帐户，以演示如何确定配置文件是否已与 PlayFab 帐户关联。 |

成功登录 PlayFab 帐户后，PlayFab 令牌将存储于 **PlayFabSettings::clientSessionTicket** 静态对象。它将用于后续 PlayFab API 调用。令牌会过期，必须每四个小时刷新一次。如果其他用户再次调用 LoginWithXbox，则会覆盖存储的令牌。令牌也可以在客户端代码中管理，通过动态替换 PlayFabSettings::clientSessionTicket，游戏可以让多个 PlayFab 用户同时玩。

# 将 Xbox 配置文件与现有 PlayFab 帐户关联

PlayFab::PlayFabClientAPI::LoginWithPlayFab 🡪 Windows::Xbox::System::User::GetTokenAndSignatureAsync 🡪 PlayFab::PlayFabClientAPI::LinkXboxAccount

PlayFab 是一个多平台解决方案，因此玩家可能已拥有 PlayFab 帐户，但尚未将帐户与其 Xbox 配置文件关联。**LinkXboxAccount** 将启用此方案，而非创建一个新的配置文件。该 **LinkXboxAccountRequest** 包含以下属性。

|  |  |
| --- | --- |
| 字段 | 描述 |
| XboxToken | 使用端点 <https://playfabapi.com> 通过 **GetTokenAndSignatureAsync** 方法生成的令牌。 |
| ForceLink | 如果 PlayFab 帐户已与其他 Xbox 配置文件关联，则将此设置为 **true** 将取消现有配置文件的关联并关联新配置文件。如果设置为 **false** 且另一个配置文件已与该帐户关联，则调用 LinkXboxAccount 将返回错误。 |

值得注意的是，传递到此 API 的请求对象不包含任何指示要关联哪个 PlayFab 帐户的字段。这是因为每个 PlayFab API 调用都要求从其中一个登录方法返回的 PlayFab 配置文件。**LoginWithPlayFab** API 允许玩家使用用户名和密码登录，但不需要 Xbox 令牌。用户登录后，可以使用 LinkXboxAccount 关联 PlayFab 配置文件。

*请务必注意，必须与 Xbox 配置文件关联方可使用所有 PlayFab 服务。若 PlayFab 帐户未与 Xbox 配置文件关联，则游戏不得允许玩家登录 PlayFab 帐户。*

# 创建一个新的 PlayFab 帐户

PlayFab::PlayFabClientAPI::RegisterPlayFabUser 🡪 Windows::Xbox::System::User::GetTokenAndSignatureAsync 🡪 PlayFab::PlayFabClientAPI::LinkXboxAccount

如果玩家的 Xbox 配置文件未关联且玩家无现有 PlayFab 帐户，则 PlayFab API 可提供几种创建新帐户的方法。创建新帐户最简单的方法是使用 **LoginWithXbox**，并将 **CreateAccount** 设置为 true。这允许您通过最少的输入以编程方式为玩家生成新帐户，但玩家可以反对在未经其本人明确输入的情况下代表其本人创建帐户。

相反，此示例演示了在玩家输入后使用 **RegisterPlayFabUser** 创建一个新帐户，然后使用上一节详述的 **LinkXboxAccount** API 将帐户关联。**RegisterPlayFabUserRequest** 对象具有以下属性。

|  |  |
| --- | --- |
| 字段 | 描述 |
| DisplayName | 与游戏中显示的 PlayFab 帐户关联的名称。长度必须介于 3 到 25 个字符之间。 |
| 密码 | 玩家想要用于登录其 PlayFab 帐户的密码。长度必须介于 6 到 100 个字符之间。 |
| UserName | 这是用于登录 PlayFab 帐户的用户名。它并不会在游戏中显示。长度必须在 3 到 20 个字符之间。 |
| RequireBothUserNameAndEmail | 如果设置为 **true**，则用户需提供用户名和电子邮件地址方可创建新帐户。如果设置为 **false**，则仅需两者其中之一即可创建帐户。 |

本列表未详尽列出与 RegisterPlayFabUserRequest 相关的属性，而仅列出了此示例中使用的属性。查看 [PlayFab 在线文档](https://api.playfab.com/documentation/client/method/LinkXboxAccount)，获取所有可用字段的完整列表。

此示例演示了如何仅使用 **UserName** 属性创建新的 PlayFab 帐户，并为 UserName 和 **DisplayName** 提供默认值，即玩家的玩家代号。这是为了最小化玩家所需的输入，并尽可能减少注册问题。

创建 PlayFab 帐户后，会自动在 PlayFabSettings 中缓存令牌，您可以开始调用 PlayFab 服务。第一个调用应为 LinkWithXbox，以立即将玩家的 Xbox 配置文件与新的 PlayFab 帐户关联。

*请务必注意，必须与 Xbox 配置文件关联方可使用所有 PlayFab 服务。若 PlayFab 帐户未与 Xbox 配置文件关联，则游戏不得允许玩家登录 PlayFab 帐户。*

# 更新历史记录

**初始发行：**2018 年 5 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/)。