

PlayFab Web 認証ブローカーのサンプル

*\* このサンプルには、2018 年 7 月の Xbox One XDK との互換性があります。*

# 概要

Xbox One で Xbox Live サービスを使用するには、プレーヤーが Xbox Live で認証を受ける必要があります。さらに、タイトルに PlayFab サービスを使用する場合は、後で Xbox Live ユーザー アカウントとリンクできる PlayFab アカウントを作成するか、取得する必要があります。PlayFab アカウントを取得する方法は多数あります。最も簡単な方法は、プレイヤーがゲームに初めてサインインしたときに、自動的に PlayFab アカウントを作成することです。この「簡単な」方法は、Xbox One ゲームの典型的なサインイン フローであり、ほとんどのゲームで推奨されるシナリオです (初めて PlayFab 認証を行う場合は、SimplePlayFabAuth サンプルで開始してください)。PlayFabWebAuthBroker サンプルでは、より高度なサインイン フローを紹介しています。これは、他の認証プロバイダーからの PlayFab アカウントの共有を有効にするゲームに適しています。特に、このサンプルでは、オープン認証プロバイダー (この場合は Facebook) でオープン認証 (OAUTH) を実行して PlayFab アカウントを取得する方法を紹介します。

# サンプルをビルドする

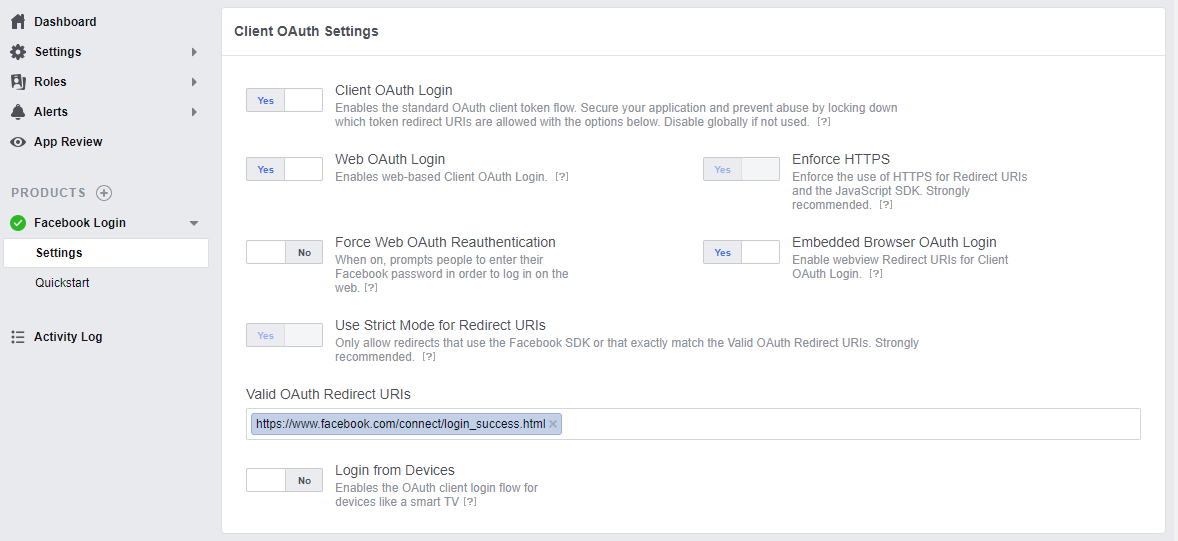
このサンプルは Visual Studio 2015 でビルドされ、2018 年 7 月の Xbox One XDK でテストされています。このサンプルは以前の XDK でも機能する場合があります

# サンプル用の Facebook アプリの構成

このサンプルでは、OAUTH プロバイダーの例として Facebook を使用します。このサンプルを使用するには、Facebook 上でアプリケーションを設定する必要があります。Facebook 上でのアプリの設定の詳細については、<https://developers.facebook.com/docs/apps> を参照してください。

## Facebook ログイン製品

アプリの設定時に、Facebook ログイン製品を追加する必要があります。製品を追加したら、有効な OAuth リダイレクト URI を指定する必要があります。 例えば：



## テスト ユーザー

サンプルから Facebook にサインインするときに使用する 1 つ以上のテスト ユーザーを作成します。テスト ユーザー アカウントは既に Facebook 上でアプリケーションを有効にするように設定されているので、サンプルを試すのに非常に便利です。

## アプリ ID の設定

サンプルを使用する前に、FacebookAuthHelper.h を編集して Facebook アプリ ID を入力する必要があります。Facebook のアプリの基本設定でアプリ ID を見つけます。次に、FacebookAuthHelper.h で「TODO」コメントを探し、「<put your Facebook App ID here>」をアプリ ID に置換します。



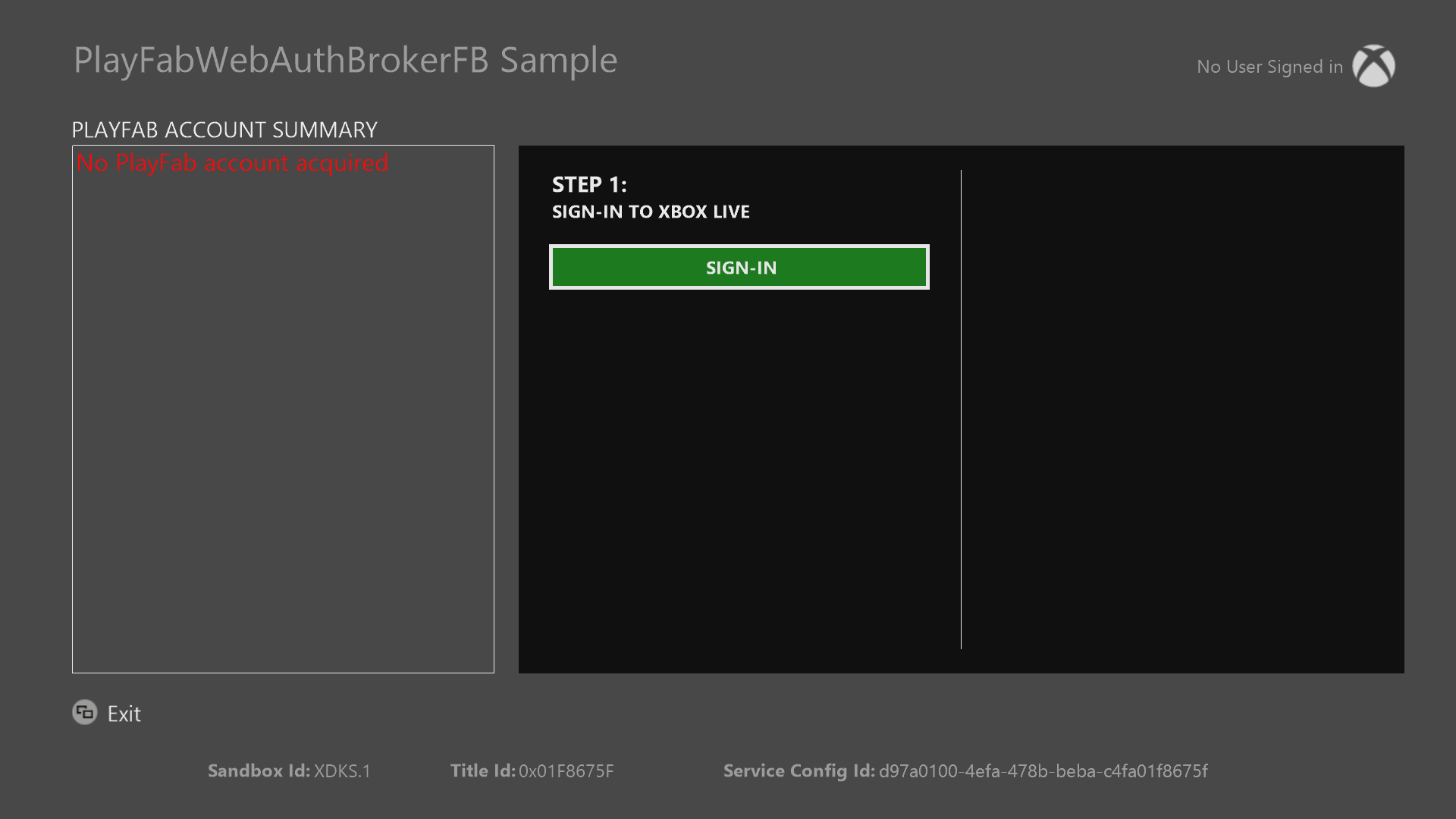
# サンプルの使用

Facebook 上ですべての設定が完了したら、Facebook アプリケーションの ID を FacebookAuthHelper.h に入力し、このサンプルを使用します。ゲームで表示される内容に似た簡単なサインイン フローが表示されます。ただし、各画面に通常はエンド ユーザーには表示されない診断情報も表示されます。サインイン フローが完了すると、アカウントの概要情報のパネルも表示されます。

## サインイン フロー

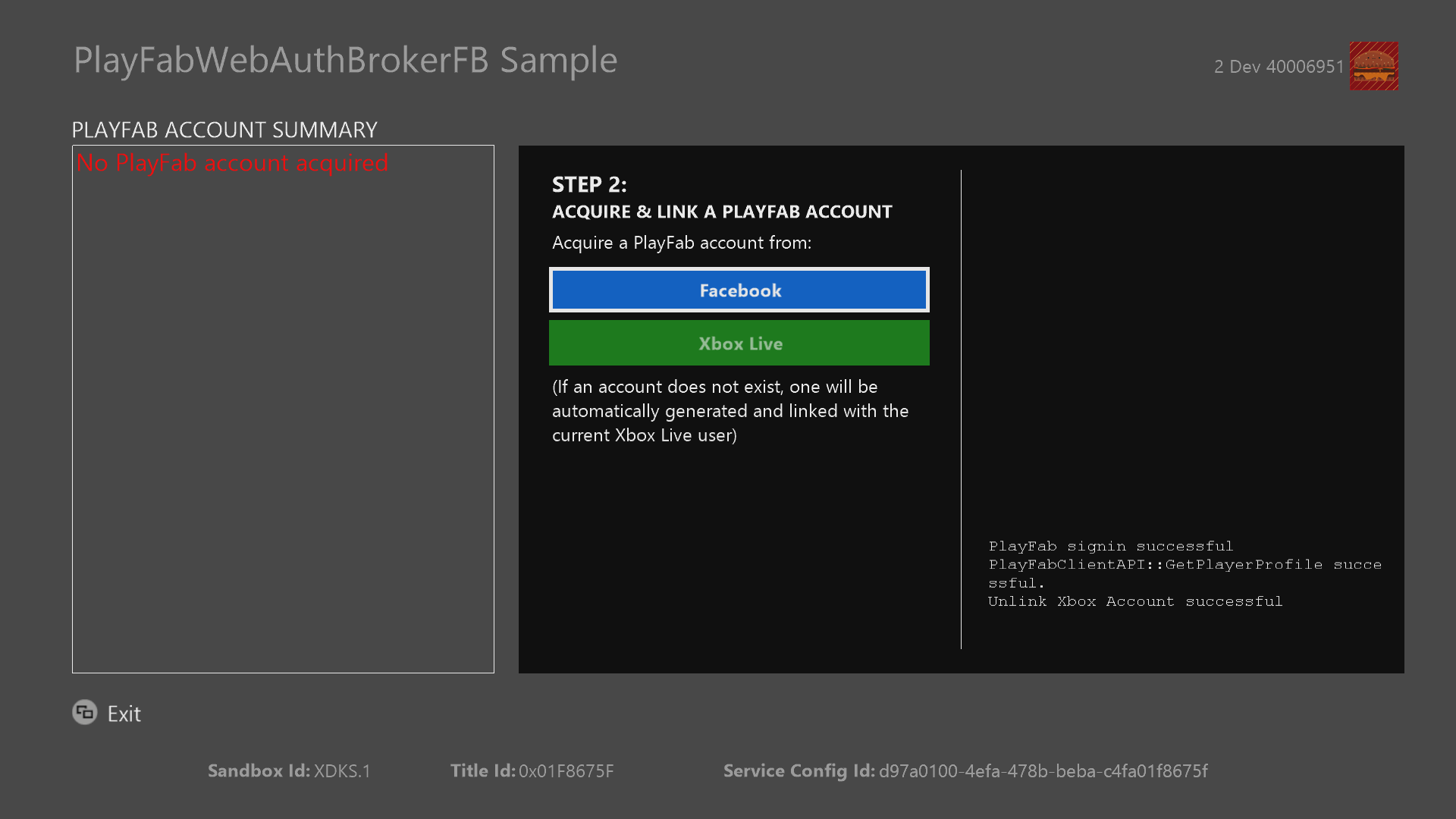
## Xbox Live サインイン

サインイン フローの最初の手順は、Xbox Live にサインインすることです。ユーザーは Xbox Live または PlayFab のサービスを呼び出す前に、必ず Xbox Live にサインインする必要があります。



## PlayFab アカウントの取得

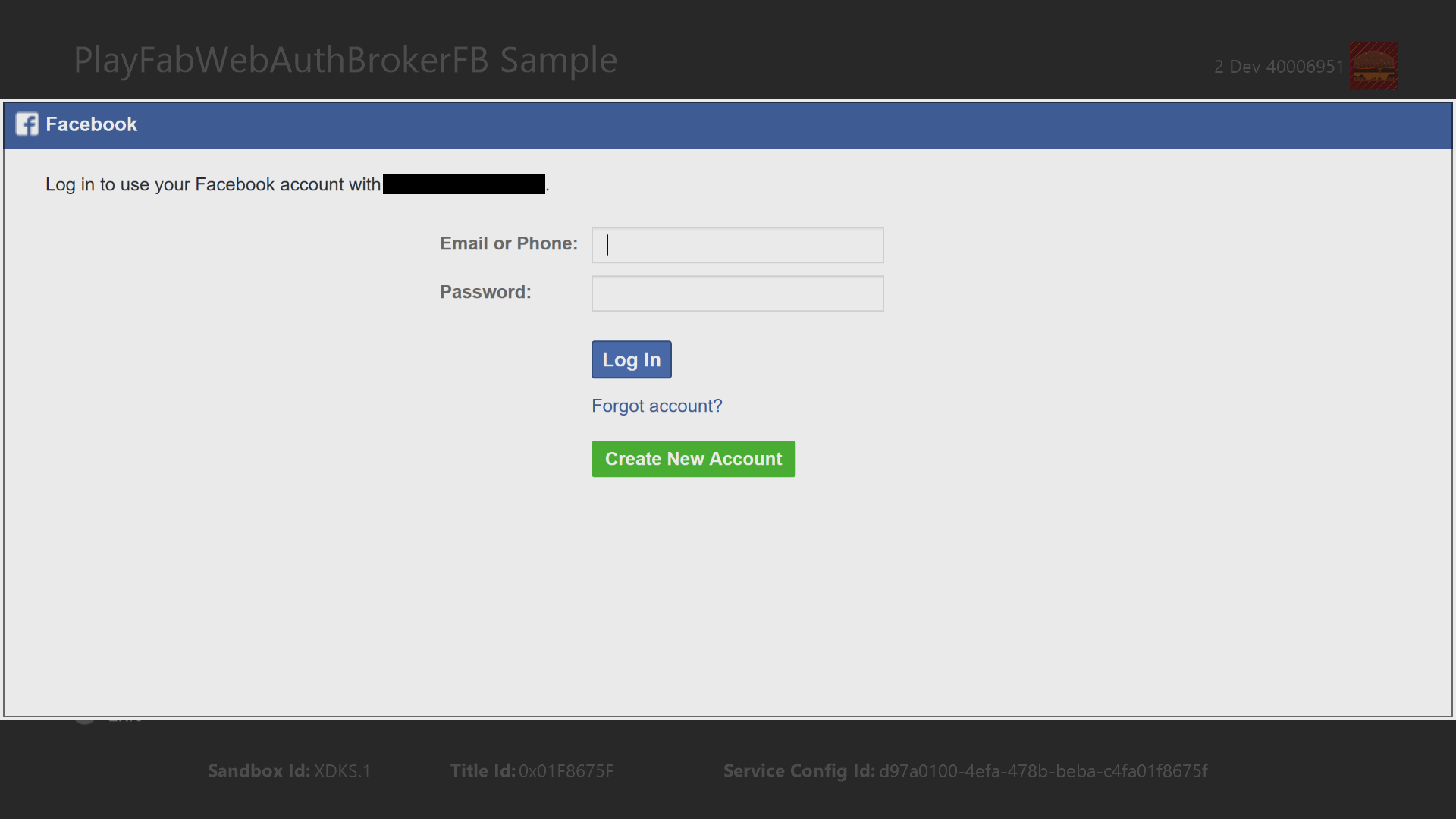
PlayFab サービスを使用するには、PlayFab アカウントを Xbox Live ユーザーとリンクする必要があります。PlayFab アカウントを作成する方法は多数あります。サインイン フローの 2 番目の手順では、ユーザーに 2 つの選択肢が提示されます。「Xbox Live」を選択した場合、このサンプルでは Xbox Live ユーザー アカウントとリンクするための既定の PlayFab アカウントだけを作成します。 「Facebook」を選択した場合、このサンプルでは Facebook などの他のログイン プロバイダーに既にリンクされている PlayFab アカウントを取得する方法を紹介します。Facebook で OAuth を実行することで、プレーヤーの Facebook アカウントと既にリンクされている PlayFab アカウントを見つけ、そのアカウントをプレーヤーの Xbox Live アカウントとリンクすることができます。



PlayFab サインイン オプションの詳細については、PlayFab のドキュメントの [Login Basics & Best Practices](https://api.playfab.com/docs/tutorials/landing-players/best-login) を参照してください

## Facebook OAuth の UI

Facebook で認証することを選択した場合、このサンプルでは Windows::Xbox::UI::SystemUI::AuthenticateAsync を呼び出します。この API 呼び出しにより、次の Web ベースのシステム UI が表示されます。



ここで、自分の Facebook アプリ用に作成したテスト ユーザー アカウントの 1 つに認証資格情報を指定します。この UI はタイトルではなくシステムによってレンダリングされます。他のシステム UI と同様に、このウィンドウではシステム キーボードの使用がサポートされていますが、テストを目的とする場合、物理的なキーボードを使用する方が便利です。

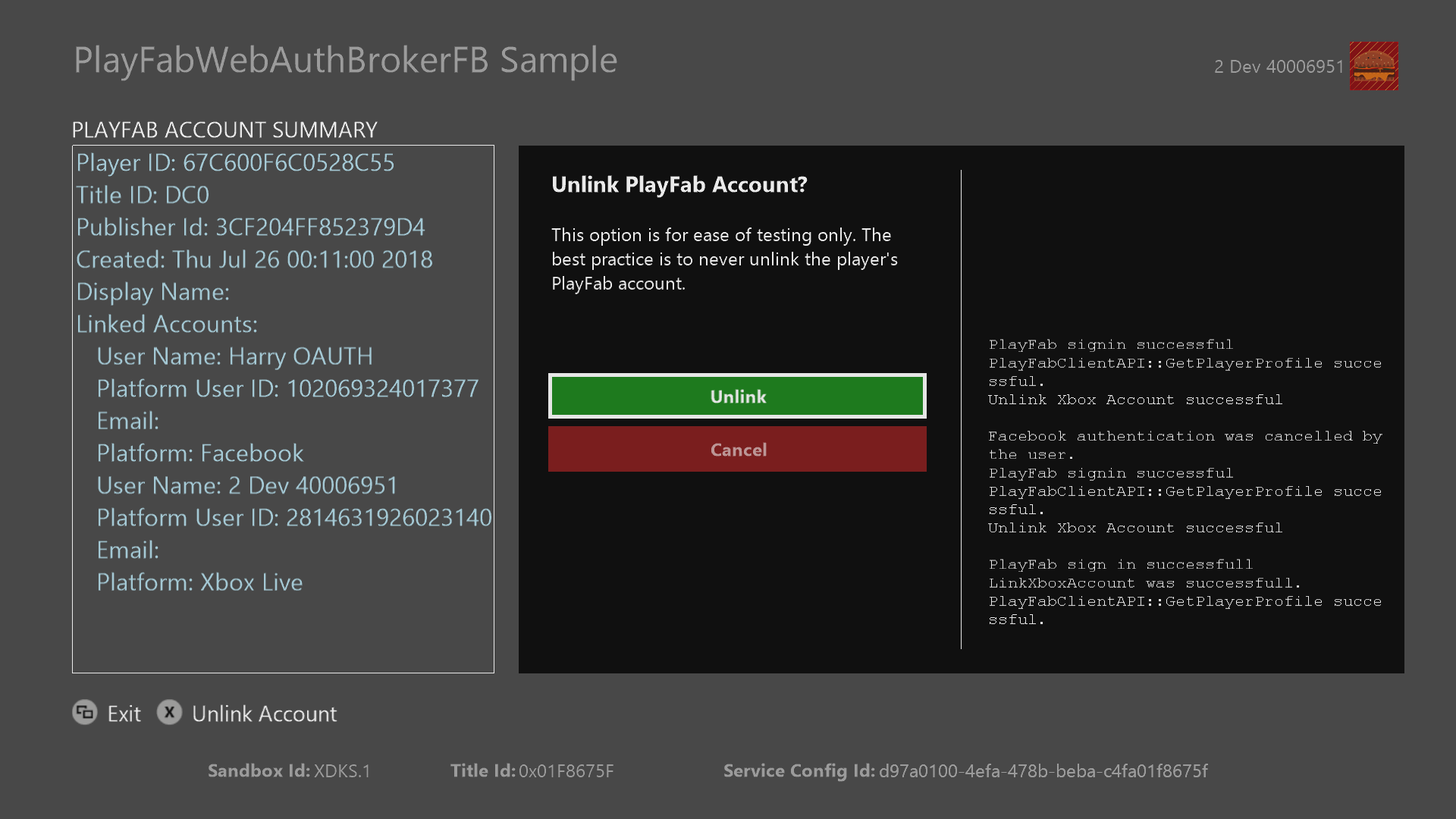
## サインインの完了

サインイン フローが完了すると、最後の画面 (手順 3) が表示されます。これで、Xbox Live と PlayFab の両方のサービスの呼び出しを開始する準備ができました。このサンプルでは、画面の左側に PlayFab アカウントの概要情報が表示されます。この情報は、PlayFabClientAPI::GetPlayerProfile の呼び出しによって表示されます。

## 

## Xbox Live アカウントから PlayFab アカウントへのリンクを解除

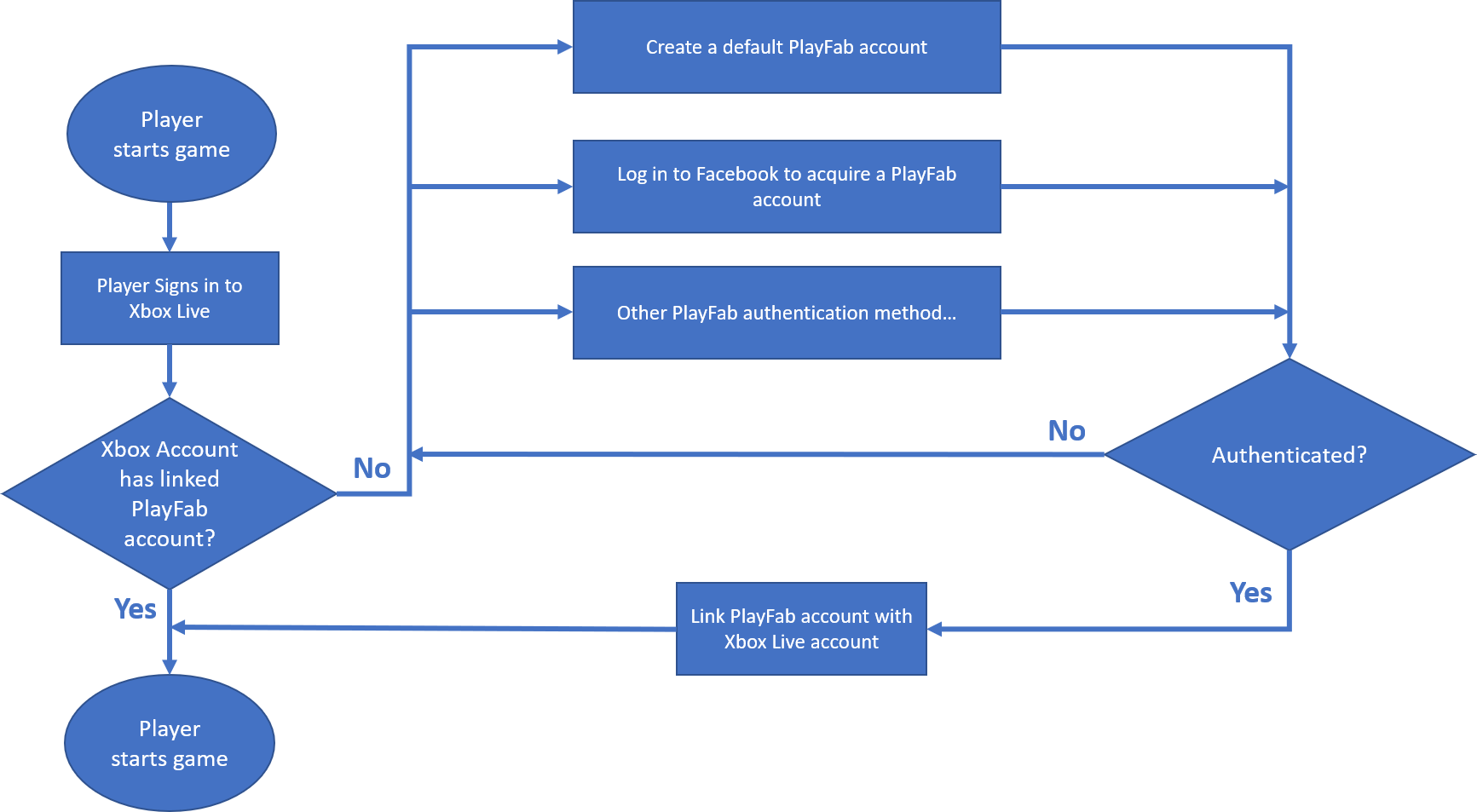
サインイン フローが正常に完了すると、このサンプルでは PlayFab アカウントと Xbox Live アカウントのリンクを解除するオプションが表示されます。PlayFab SDK によって、この目的のための API (PlayFabClientAPI::UnlinkXboxAccount) が提供されます。ただし、ベスト プラクティスとして、通常はプレーヤーの PlayFab アカウントのリンクを解除しないことをお勧めします。アカウントのリンクを解除すると、プレーヤー用に維持されている PlayFab サーバー側の状態が失われる可能性があります。さらに、同じアカウントへのリンクは、PlayFab の同じ発行者 ID を共有する複数のタイトル間で正常に共有できます。アカウントのリンクを解除すると、その結果、広範囲にわたって破壊がもたらされ、プレイヤーが混乱する可能性があります。 このため、このサンプルではテストを目的としたリンク解除機能のみが提供されます。



# 実装上の注意

## 高度なサインイン フロー

このサンプルでは、次のサインイン フローを実装しています。



これは、PlayFab のドキュメントの [Login Basics & Best Practices](https://api.playfab.com/docs/tutorials/landing-players/best-login) に記載されているフローチャートの簡易版です。このサンプルで実装されているサインイン フローは、プレーヤーが常に最初の手順として Xbox Live でサインインすることを前提としているため、Xbox One のタイトルにも適しています。論理サインイン フローは PlayFabWebAuthBrokerFB.h/.cpp で実装されます。

**PlayFab アカウントのリンク**

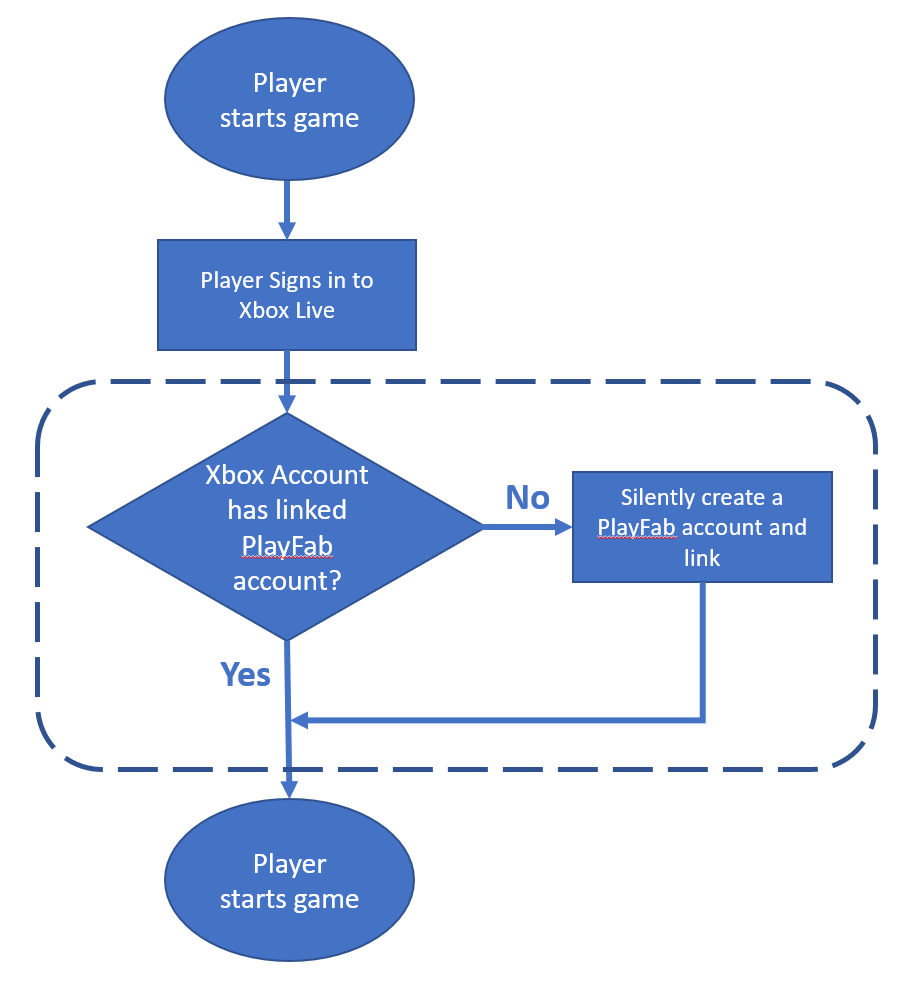
高度なサインイン フローを実装する場合、PlayFab アカウントを Xbox Live アカウントにリンクするときに常に注意が必要です。Xbox Live アカウントが既に PlayFab アカウントにリンクされている場合、別の PlayFab アカウントとの「リンクを強制」すると、元のアカウントは破棄されます。これにより、元の PlayFab アカウントに関連する進行状況やその他のアカウント状態が失われると、プレーヤーが混乱し、不満を感じることがあります。ここに示すサインイン フローでは、他の選択肢をプレイヤーに提示する前に PlayFab アカウントの存在をテストするため、そのような問題を回避することができます。具体的には:

* PlayFabClientAPI::LoginWithXbox のエラー ハンドラーで、PlayFabErrorCode::PlayFabErrorAccountNotFound がチェックされます。このエラーの存在によって、Xbox Live アカウントに既にリンクされている PlayFab アカウントがないことがサンプルのロジックに通知されます。
* PlayFabClientAPI::LinkXboxAccountRequest では、サンプルで ForceLink オプションが ***false*** に設定されます。推奨されるサインイン フローに従う場合は、このオプションを true に設定する理由はないはずです。

既存の PlayFab アカウントを不必要に放棄するのを避けるために、*ForceLink を true に設定する際には慎重に行うこと*をお勧めします。

**単純なサインイン フロー**

次の図は、最小限のサインイン フローを示しています。



簡略化されたサインイン フローは最小限ですが、それでも唯一の認証プロバイダーとして Xbox Live に依存しているほとんどの Xbox One ゲームに適しています。追加の UI フローをプレイヤーに表示することなく、破線の範囲内のすべてを*自動的に*実行できます。特に、LoginWithXboxRequest 構造に入力する際に CreateAccount = true を指定して、PlayFab アカウントがまだ存在してない場合に新しいアカウントを作成するように指定します。ほとんどのゲームで、この方法が推奨されます。

## OAuth の実装

Facebook OAuth の実装は、FacebookAuthHelper と呼ばれるスタンドアロン クラスで行えます。これは、「WebAuthBrokerFB」と呼ばれる別のサンプルの一部である「FacebookHelper」の簡易版です。Facebook とは異なる認証プロバイダーに対して OAuth を実行する場合は、準備の例として FacebookAuthHelper から始めて、必要に応じて変更します。

# 既知の問題点

## Windows::Xbox::UI::SystemUI::AuthenticateAsync

* ダイアログが応答しなくなり、物理的なキーボード入力を受け付けなくなることがあります。サンプルを再起動することによる回避策
* 正常にログインした後、その後の呼び出しでダイアログが表示されない場合がありますが、非同期操作は正常に完了します (おそらく、前回の正常なログインからキャッシュされたログイン結果が使用される)。
* ログインが成功するたびに、API によって例外がスローされます。例外が処理され、無視できますが、開発者にとっては目障りです。

# 更新履歴

サンプルの初回リリース 2018 年 11 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。