

XSTS 2018 サーバーサンプル

# 概要

カスタムWebサービスを Xbox Live 対応のタイトルと統合するには、認証とユーザー識別に XSTS トークンを使用する必要があります。 クライアントからトークンを取得するのは、GetTokenAndSignatureAsync() を使用するのは簡単ですが、XSTS トークンと統合して操作することは一部の開発者にとっては難しい場合があります。 このサンプルは、XSTS トークンの復号化と検証、およびユーザーに代わって認証された呼び出しを行うために、必要なサービスとの通信につかわれる、サンプルコード（バックエンドからバックエンドまたはb2b）を持つ、機能する Web サーバーです。

これは、独自のトークン処理を別の言語で書く場合、教育用の例として使用することも、Webサービスを構築するための基本フレームワークとして使用することもできます。

# サンプルをビルドする

このサンプルは自己完結型ですが、ソリューションを最初に開いて構築するときにダウンロードする必要がある NuGet パッケージ（Newtonsoft.Json、Jose.JWT、および Azure Key Vault との接続に必要なパッケージ）に依存しています。 サンプルをコンパイルするには、Xsts2018.sln を開いてからコンパイルするだけです。

これらをプロジェクトの条件付きコンパイル シンボル テキスト ボックス（[プロジェクトのプロパティ] - > [ビルド] - > [すべての構成]の下にあります）に追加することで、以下を有効にできます。

* LINUX - リバースプロキシからトラフィックが入ってきて、これらを追加する Linux で一般的に使用されているで FWD ヘッダを有効にします。
* XSTS\_LEGACY - XSTS 2017形式および Symmetric Shared Key Relying Party ではなく、レガシー XSTS トークン（依存パーティ証明書を含む非対称ドラフト 7 JWE）の処理を有効にします。

# サンプルの使用

Azure サービスを構成し、それと共に実行するようにサンプルを展開する方法については、サンプルに付属の XSTS 2018 サンプルクイックスタートガイドを参照してください。

# 実装上の注意

このサンプルには、次の主な機能とデザインがあります。

* 生のソースコードまたは公開されているオープンソース ソフトウェア ソリューション （Newtonsoft.Json、Jose.JWT、.NET Core 2.1）を使用した、すべてのトークン処理クラスおよび関数へのソースコードアクセス
* Web サービスを作成するための出発点として使用できるサンプル
* C# ASP.NET Core 2.1 を使用して複数のサーバー OS 上で実行する機能
  + Windows Server 2016 を検証済み
  + Linux（Ubuntu 16.04） を検証済み
* レガシー XSTS トークン（XDP 構成 JWT Draft 7 Asymmetric ）と新しい 2017 XSTS トークン（UDC 構成 JWT RFC 対称共有鍵）の両方のトークン処理と検証
* Xbox Live サーバー間要求に必要なサービストークンの取得と管理
* ユーザーに代わって Xbox Live サーバー間要求に必要な代理認証 XSTS トークンの取得をして使用
* 証明書利用者向けの開発センター、ビジネスパートナー証明書、およびNSALセットアップ用のデベロッパーセンターのステップを追った構成手順。

# 既知の問題点

* なし

# 更新履歴

2018 年 10 月 26 日–v1.0

# プライバシーステートメント

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。