

Custom Service ハンズオン

目次

1. 全体像の確認
2. 事前準備
3. Custom Service 開発

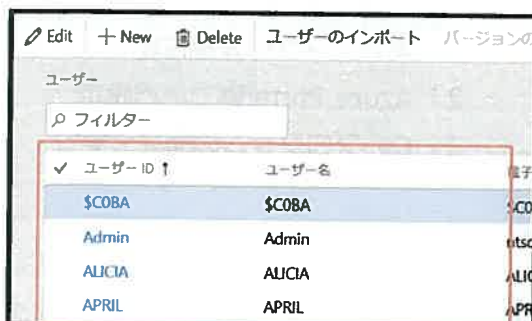
2. 事前準備

2. 事前準備

C#からD365FOのプログラムを実行するためには、**D365FOに実行ユーザーが登録されていること**
Azure Portal側からの承認設定が必要です。

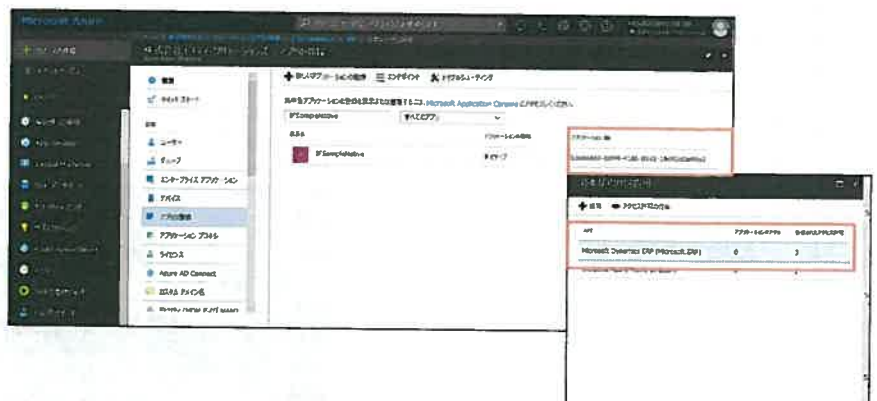
具体的には、ユーザーID、パスワード、アプリケーションキーが必要となります。

D365FOにユーザーが登録されていること
 システム管理>ユーザー>ユーザー



ユーザー ID ↑	ユーザー名
\$COBA	\$COBA
Admin	Admin
AUCIA	AUCIA
APRIL	APRIL

Azure Portal側からアクセスユーザーに承認設定がされていること
 システム管理>ユーザー>ユーザー

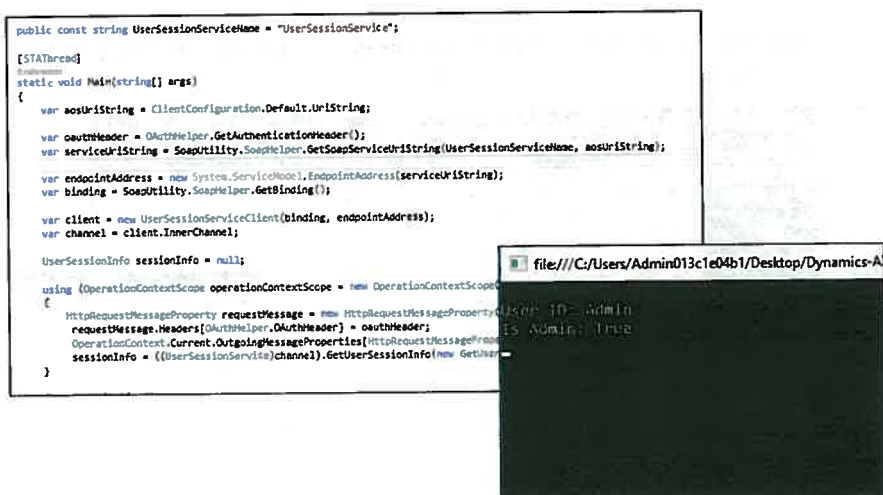


3. Custom Service 開発

3. Custom Service 開発

C#からD365F0のプログラムを実行する開発方法を学習します。

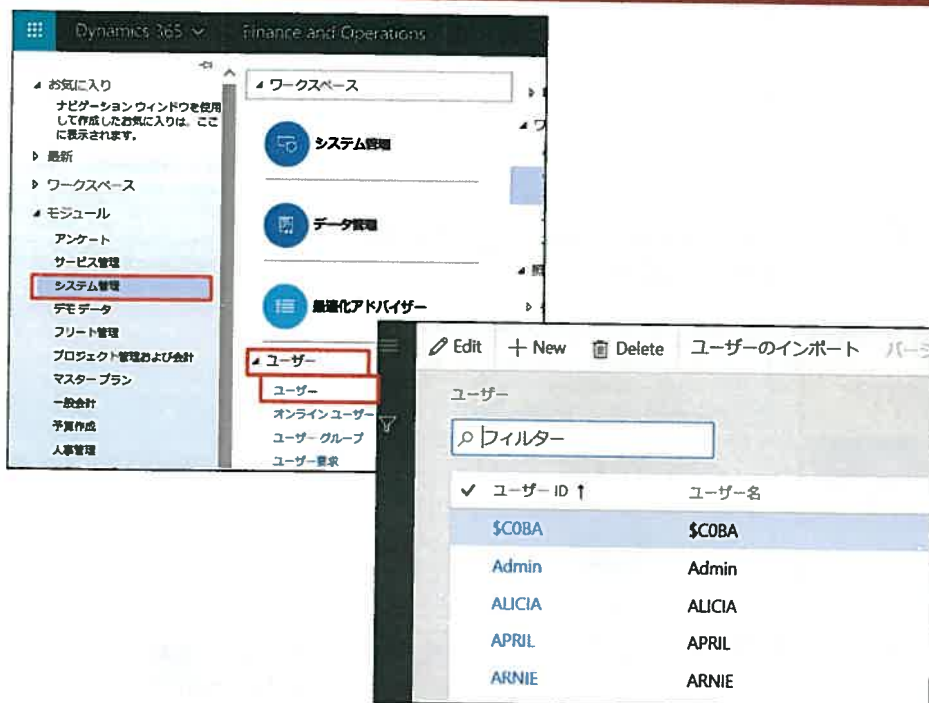
開発は以下の手順となります。



<開発手順>

- 1) Custom Serviceの
実行ユーザー登録確認
- 2) Azure Portal側で承認設定
- 3) C# プログラムの開発・実行

3. Custom Service 開発



Step 1

D365FOにサインイン。

Step 2

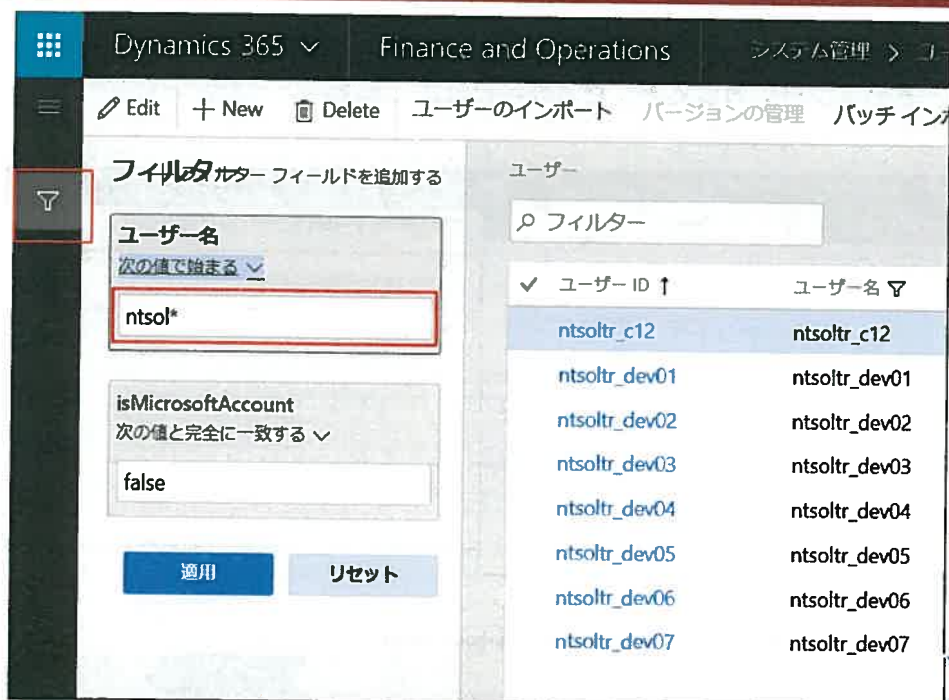
システム管理>ユーザー>ユーザーを押下。

1)

2)

3)

3. Custom Service 開発



Step 3

フィルターボタンを押下。

Step 4

ユーザー名から実行ユーザー名を入力。

<確認>

実行ユーザーが登録されていることを確認。

1)

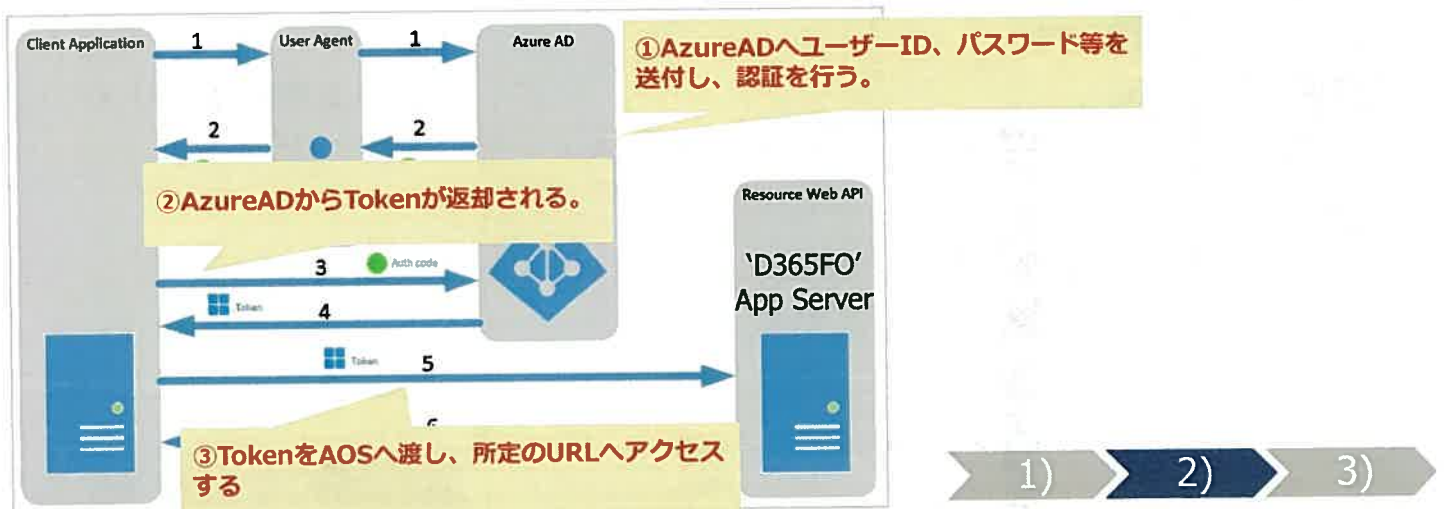
2)

3)

3. Custom Service 開発

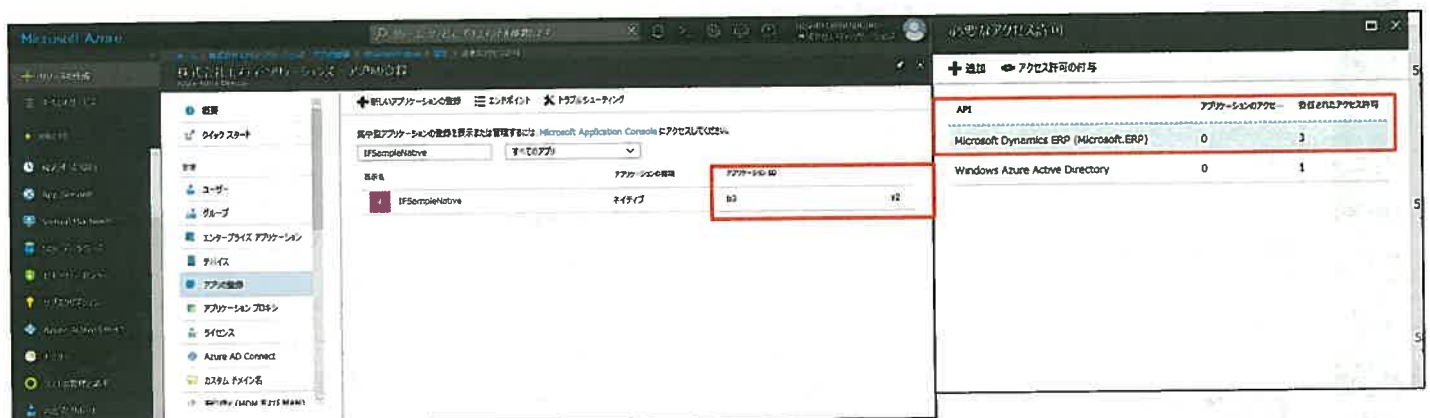
C#からOperationsのプログラムを実行するためには、Azure側からの承認設定が必要です。

具体的には、ユーザーID、パスワード、アプリケーションキー、(ODataの場合はシークレットキー)が必要になります



3. Custom Service 開発

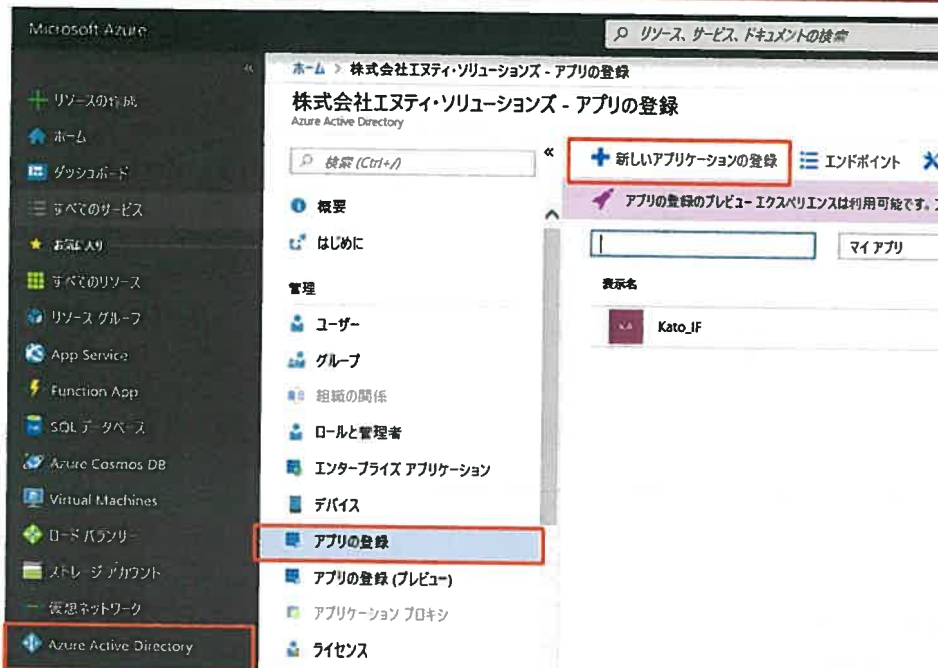
Azure PortalからAzure Active Directoryの設定よりアプリ登録を行い、アプリケーションIDの発行、および、そのアプリに対してアクセス許可を設定します。



参考URL :

<https://docs.microsoft.com/en-us/dynamics365/unified-operations/dev-itpro/data-entities/services-home-page>

3. Custom Service 開発



Step 1

Azure Portalへサインイン。
<https://portal.azure.com/>

Step 2

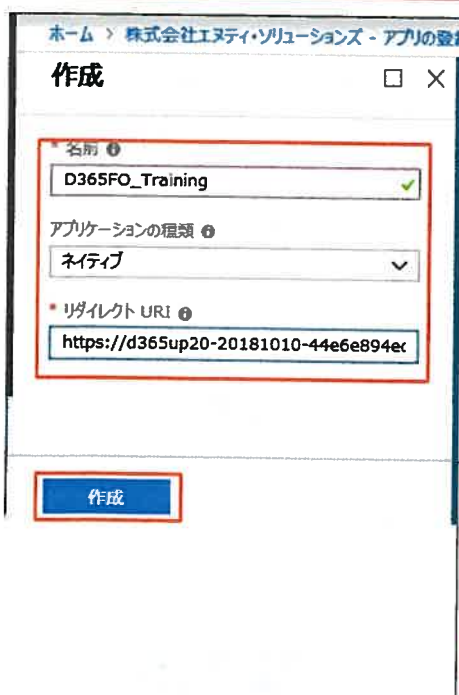
Azure Active Directory> アプリ登録 を押下。

Step 3

新しいアプリケーションの登録を押下。

1) 2) 3)

3. Custom Service 開発



Step 4

入力項目に値を設定。
名前：任意の名称
アプリケーションの種類：ネイティブ
リダイレクトURL：D365FOのURL

Step 5

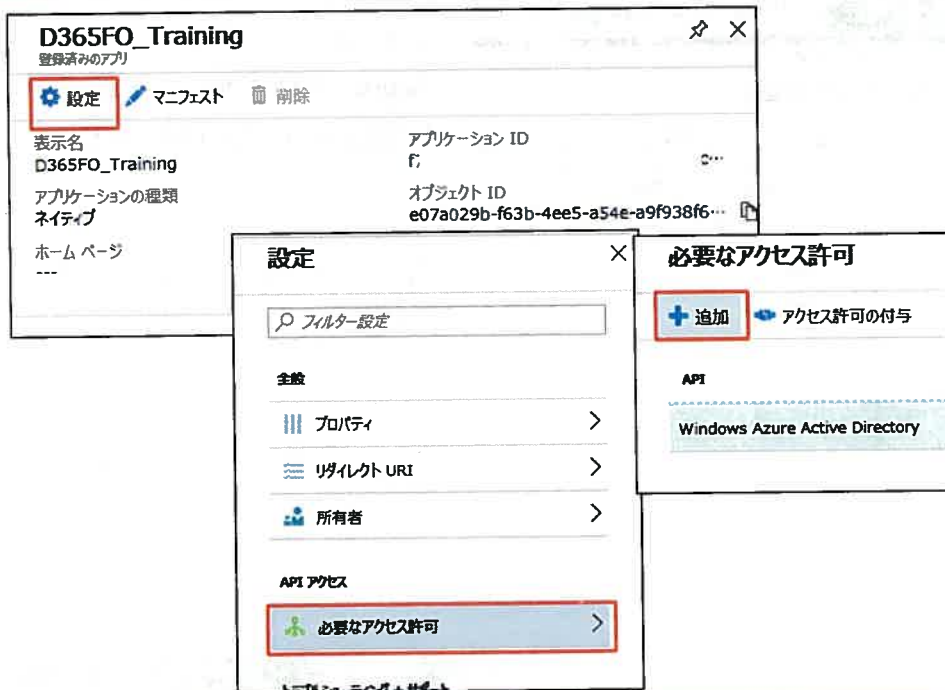
[作成]ボタンを押下。

<確認>

作成したアプリケーションが追加されていることを確認。

1) 2) 3)

3. Custom Service 開発



Step 6

[設定]ボタンを押下。

Step 7

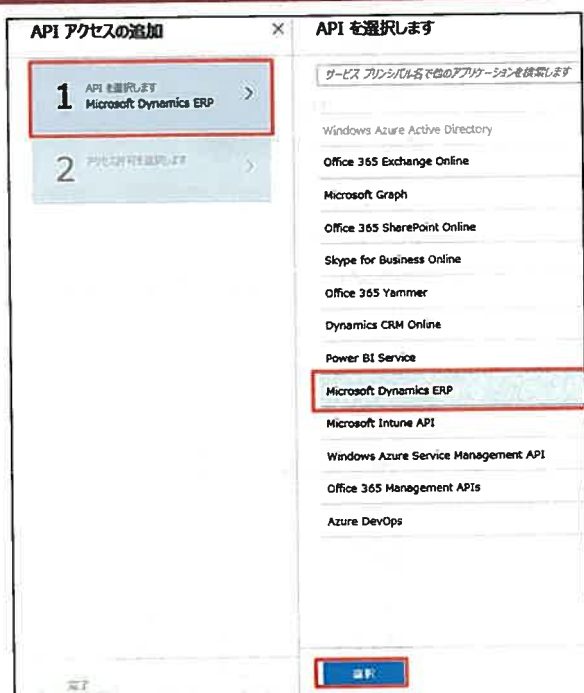
[必要なアクセス許可]を押下。

Step 8

[追加]ボタンを押下。



3. Custom Service 開発



Step 9

[1 APIを選択します]を押下。

Step 10

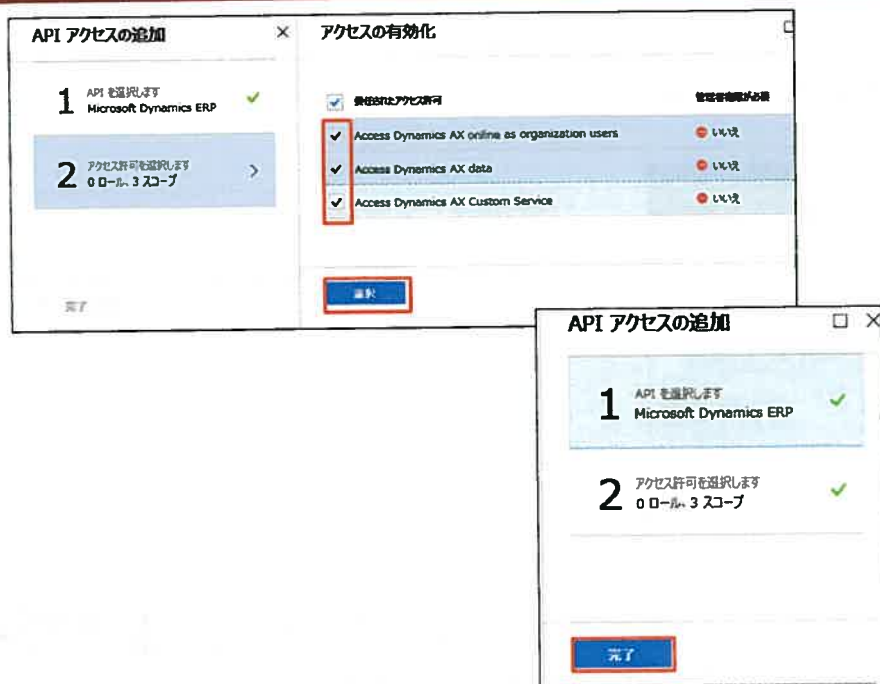
[Microsoft Dynamics ERP]を選択。

Step 11

[選択]ボタンを押下。



3. Custom Service 開発



Step 12

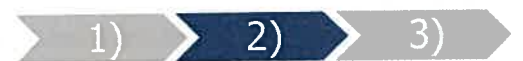
すべてにチェックを付与。

Step 13

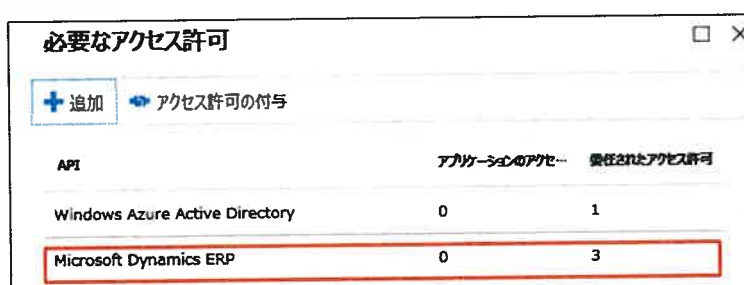
[選択]ボタンを押下。

Step 14

[完了]ボタンを押下。



3. Custom Service 開発



<確認>

Microsoft Dynamics ERP が追加されたことを確認。

Step 15

[アクセス許可の付与]を選択。

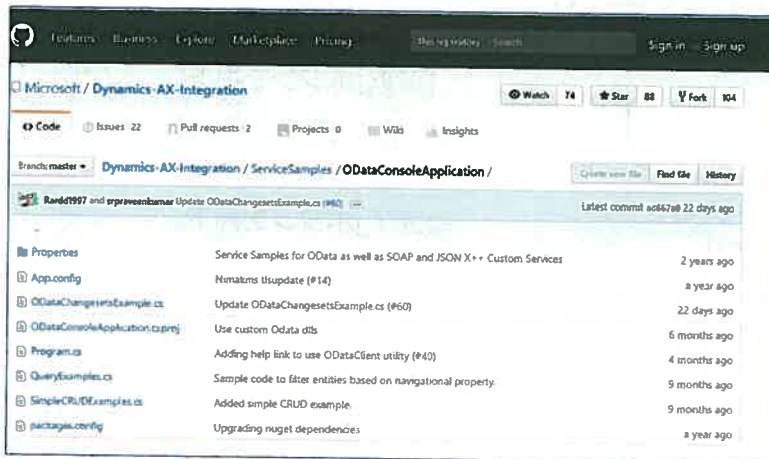
Step 16

[はい]ボタンを押下。



3. Custom Service 開発

Microsoft社より外部接続用のC#のプログラムサンプルが公開されています。
本トレーニングではこちらを利用し、開発を行います。

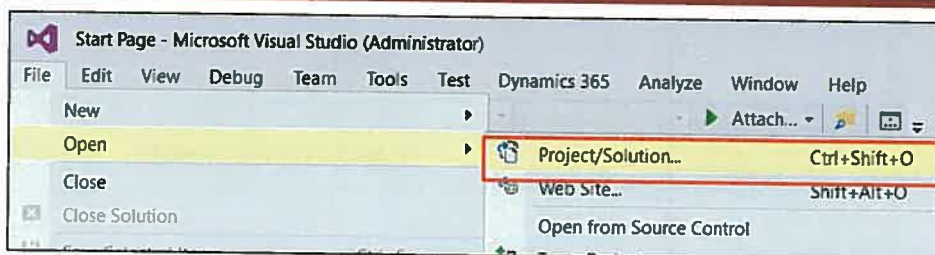


参考URL：

<https://github.com/Microsoft/Dynamics-AX-Integration/tree/master/ServiceSamples/ODataConsoleApplication>



3. Custom Service 開発



Step 1

Visual Studioを管理者権限で起動。

Step 2

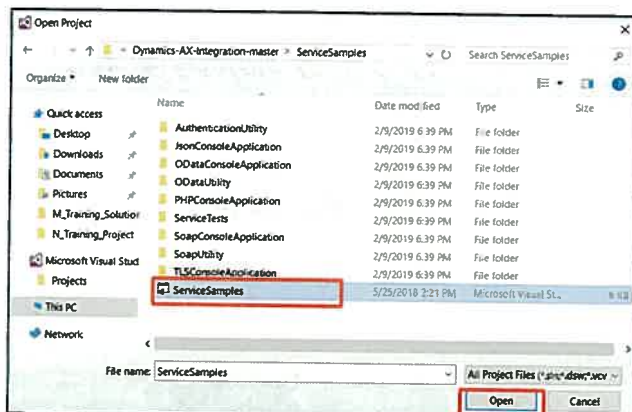
File> Open>
Project/Solution..を選択。

Step 3

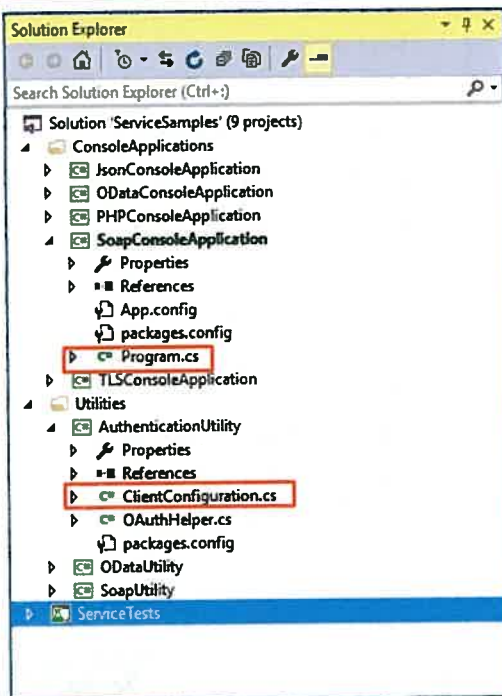
Service Samplesを選択。

Step 4

[開く]ボタンを選択。



3. Custom Service 開発



本トレーニングでは、以下のクラスを編集します。

- SoapConsoleApplication > Program.cs
⇒メイン処理が実行されるプログラム。
- AuthenticationUtility > ClientConfiguration.cs
⇒接続先情報を保持するプログラム。

1) 2) 3)

3. Custom Service 開発

```
namespace AuthenticationUtility
{
    24 references
    public partial class ClientConfiguration
    {
        16 references | 0/2 passing
        public static ClientConfiguration Default { get { return ClientConfiguration.OneBox; } }

        public static ClientConfiguration OneBox = new ClientConfiguration()
        {
            UriString = "https://survey1f92ec383263177127a0s.cloudapp.dynamics.com/",
            UserName = "nts0101@nts01.onmicrosoft.com",
            Password = "",
            //SOAP
            ActiveDirectoryClientAppId = "",
            ActiveDirectoryClientAppSecret = "",
            ActiveDirectoryResource = "https://survey1f92ec383263177127a0s.cloudapp.dynamics.com",
            ActiveDirectoryTenant = "https://login.microsoftonline.com/nts01.onmicrosoft.com",

            //OData
            //ActiveDirectoryClientAppId = "f6880549-8203-4fe6-8107-b2eb4e6fc618",
            //ActiveDirectoryClientAppSecret = "IqQKmFE2rH/Hjagbtp/5kIB1AkZc/ktCfLnKUIz3hKso=",

            // Change TLS version of HTTP request from the client here
            // Ex: TLSVersion = "1.2"
            // Leave it empty if want to use the default version
            TLSVersion = "",
        };
    }
}
```

Step 5

ClientConfiguration.csより、接続先情報を書き換えます。

- 接続先情報 (UriString)
- ユーザーID (UserName)
- パスワード (Password)
- AzureActiveDirectoryのアドレス (ActiveDirectoryResource)
- アプリケーションキー (ActiveDirectoryClientAppId)

1) 2) 3)

3. Custom Service 開発

```
public const string UserSessionServiceName = "UserSessionService";

[STAThread]
// references
static void Main(string[] args)
{
    var aosUriString = ClientConfiguration.Default.UriString;

    var oAuthHeader = OAuthHelper.GetAuthenticationHeader();
    var serviceUriString = SoapUtility.SoaHelper.GetSoapServiceUriString(UserSessionServiceName, aosUriString);

    var endpointAddress = new System.ServiceModel.EndpointAddress(serviceUriString);
    var binding = SoapUtility.SoaHelper.GetBinding();

    var client = new UserSessionServiceClient(binding, endpointAddress);
    var channel = client.InnerChannel;

    UserSessionInfo sessionInfo = null;

    using (OperationContextScope operationContextScope = new OperationContextScope(channel))
    {
        HttpRequestMessageProperty requestMessage = new HttpRequestMessageProperty();
        requestMessage.Headers[OAuthHelper.OAuthHeader] = oAuthHeader;
        OperationContext.Current.OutgoingMessageProperties[HttpRequestMessageProperty.Name] = requestMessage;
        sessionInfo = ((UserSessionService)channel).GetUserSessionInfo(new GetUserSessionInfo()).result;
    }
}
```

Step 6

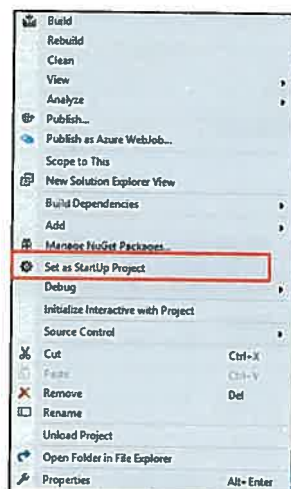
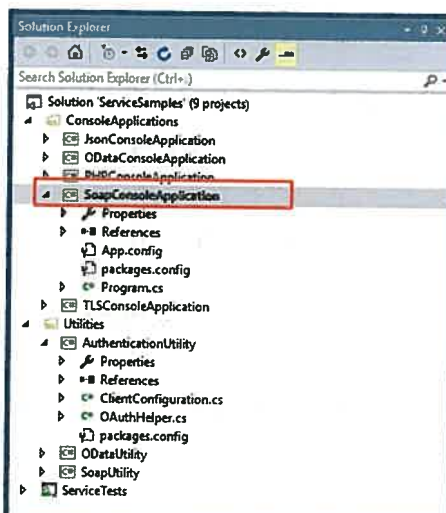
Program.csより、
どのサービス（X++のプログラム）を呼び出すか接続先情報を書き換えます。

<補足>

変数(UserSessionServiceName)に指定する値は、D365FOのサービスグループ名となります。

1) 2) 3)

3. Custom Service 開発



Step 7

SoapConsoleApplicationを
右クリック > Set as StartUp
Objectを選択。

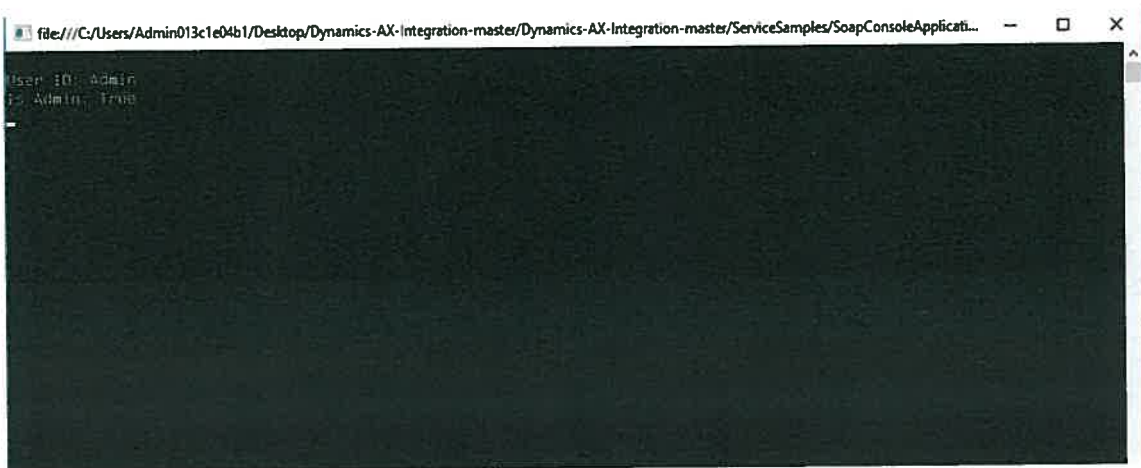
Step 8

Startボタン(または[F5]キー)
よりプログラムを実行。

1) 2) 3)

3. Custom Service 開発

Program.csで指定したサービスが実行され、コンソール画面に結果が表示されることを確認します。



1)

2)

3)

