

Tworzymy nowe rozwiązanie dla Sfery nexo w MS Visual Studio 2017 wykorzystujące C# i Windows Forms

Copyright © 2020 InsERT S.A.

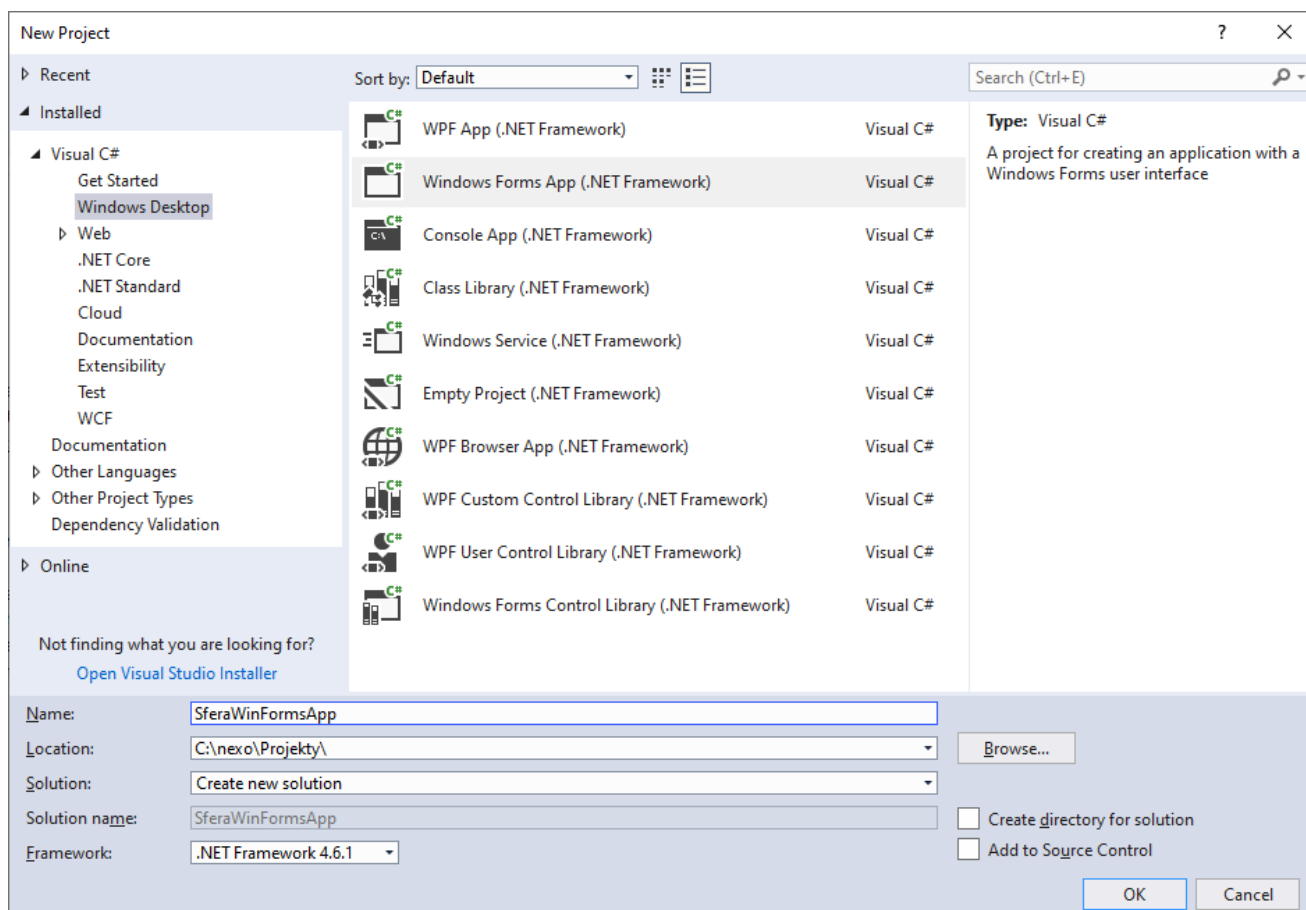
Opisujemy jak krok po kroku utworzyć najprostszą aplikację korzystającą ze Sfery nexo realizującą logowanie przez użytkownika do podmiotu i wykonującą prostą operację z użyciem Sfery. Sposób wdrożenia takiego rozwiązania omówiono w osobnym dokumencie.

WYMAGANIA

- Zainstalowany Visual Studio 2017 z C#, Windows Forms, .NET Framework 4.6.1
- Zainstalowane nexo
- Dostęp do podmiotu nexo działającego na MS SQL Server z licencją Subiekta nexo PRO (w przykładzie użyto demonstracyjnego podmiotu Subiekta nexo na lokalnym serwerze SQL)
- Zainstalowane nexo SDK w wersji zgodnej z wersją podmiotu (w przykładzie użyto wersji 29.0.0.3388)

KROK 1. TWORZYMY NOWY PROJEKT

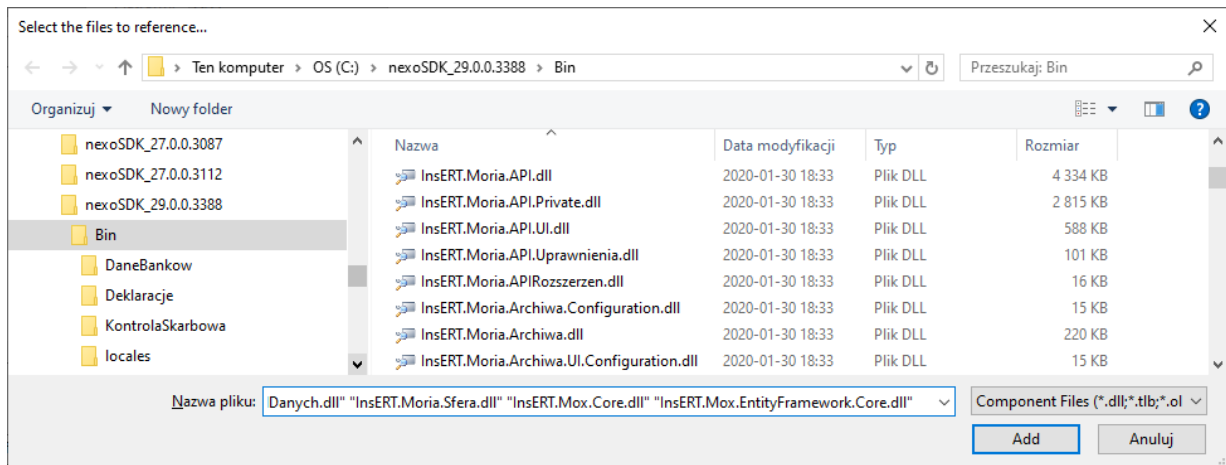
1. Uruchamiamy Visual Studio 2017
2. Wybieramy z menu opcję **File – New – Project**, a następnie w okienku dodawania projektu:
3. Wybieramy **Visual C# – Windows Desktop – Windows Forms App (.NET Framework)**
4. Wypełniamy pole **Name: SferaWinFormsApp**
5. Wybieramy **Framework: .NET Framework 4.6.1**



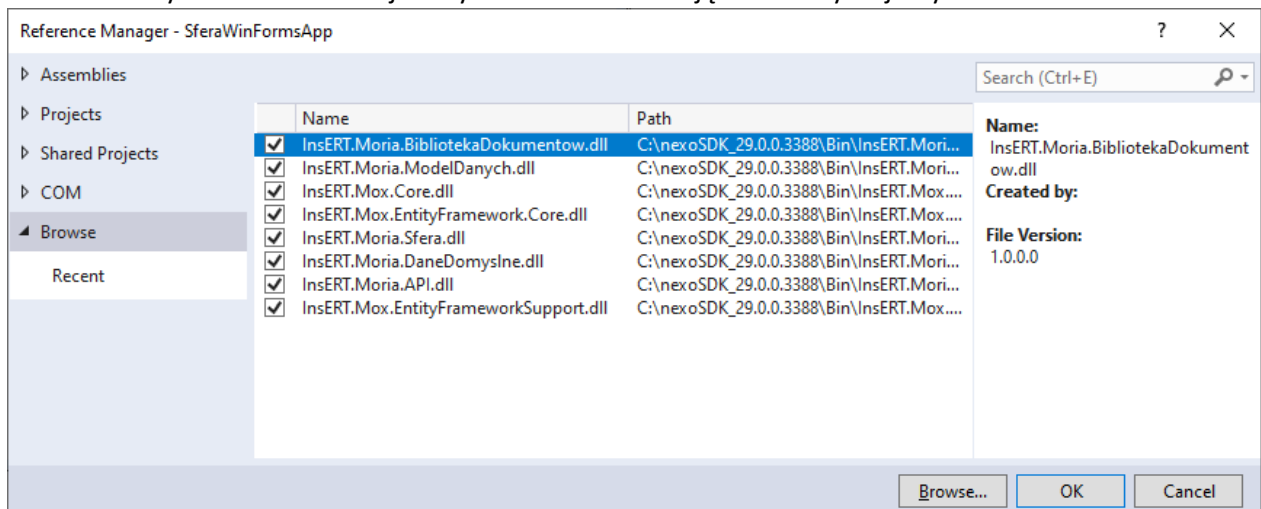
6. Zatwierdzamy wciskając przycisk **OK** i w efekcie zostaje wygenerowany projekt z podanymi parametrami.

KROK 2. DODAJEMY POTRZEBNE REFERENCJE DO PROJEKTU

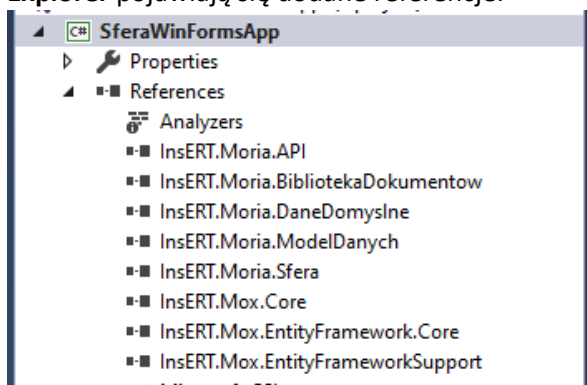
1. Wybieramy z menu opcję **Project – Add Reference...**, a następnie w oknie **Reference Managera**:
2. Wciskamy przycisk **Browse...** i w okienku nawigujemy do wnętrza folderu nexo **SDK\Bin**, np. **C:\nexoSDK_29.0.0.3388\Bin**
3. W nazwę pliku do dodania wklejamy listę potrzebnych bibliotek skopiowaną z pomocy do Sfery nexo:
"InsERT.Mox.EntityFrameworkSupport.dll" "InsERT.Moria.API.dll"
"InsERT.Moria.BibliotekaDokumentow.dll" "InsERT.Moria.DaneDomyslne.dll"
"InsERT.Moria.ModelDanych.dll" "InsERT.Moria.Sfera.dll" "InsERT.Mox.Core.dll"
"InsERT.Mox.EntityFramework.Core.dll"



4. Potwierdzamy dodanie referencji do tych bibliotek wciskając **Add** i uzyskujemy taki efekt



5. Kończymy dodawanie referencji zamykając okno **Reference Managera** wciskając **OK**. W oknie **Solution Explorer** pojawiają się dodane referencje.



KROK 3. DODAJEMY KOD URUCHAMIAJĄCY SFERĘ DO KLASY PROGRAM

1. Korzystając z okna **Solution Explorer** otwieramy plik **Program.cs** dwuklikając na jego nazwie.
2. Do klasy **Program** dodajemy metodę uruchamiającą Sferę, ale bez logowania, do celów testowych pozostawiając możliwość połączenia z podanym serwerem i bazą (tutaj: serwer na maszynie lokalnej i baza nexo_Sub29.3388):

```
private static Uchwył UruchomSfere()
{
    DanePolaczenia danePolaczenia = null;

    string pName = "/UruchomionePrzezInsLauncher";
    if (Environment.GetCommandLineArgs().Any(a => string.Compare(a, pName, true) == 0))
        danePolaczenia = DanePolaczenia.Odbierz();
    else
        danePolaczenia = DanePolaczenia.Jawne("(local)", "nexo_Sub29.3388", true);
    MenedzerPolaczen mp = new MenedzerPolaczen();
    return mp.Polacz(danePolaczenia, ProductId.Subiekt);
}
```

3. Do klasy **Program** dodajemy właściwość przechowującą uchwyt do Sfery:

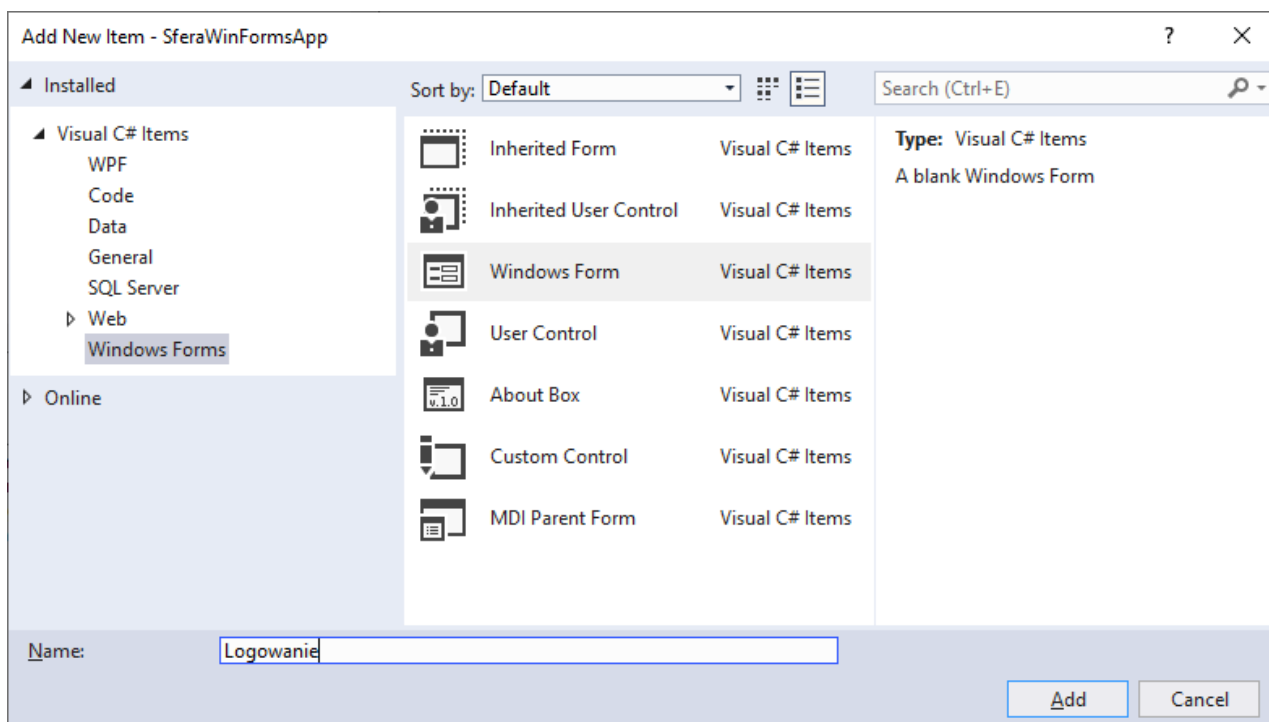
```
private static Uchwył _sfera = null;
public static Uchwył Sfera
{
    get
    {
        if (_sfera == null)
            _sfera = UruchomSfere();
        return _sfera;
    }
}
```

4. Na początku pliku **Program.cs** dodajemy:

```
using InsERT.Moria.Sfera;
using InsERT.Mox.Product;
```

KROK 3. TWORZYMY OKNO LOGOWANIA UŻYTKOWNIKA

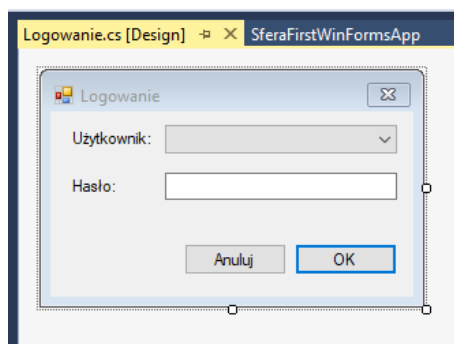
1. Dodajemy nowe okno naszego rozwiązania wybierając z menu opcję **Project – Add Windows Form**
2. W oknie wybieramy **Visual C# Items – Windows Forms** i dalej **Windows Form**
3. W pole **Name** wpisujemy **Logowanie**
4. Zanim zatwierdzimy klawiszem **Add** mamy taki efekt:



Na nowododanej formatce **Logowanie**, za pomocą okna **Toolbox** dodajemy nowe kontrolki przeciągając je na formatkę w odpowiednie miejsce, a następnie za pomocą okna **Properties** ustawiamy ich właściwości jak poniżej:

1. Dodajemy **Label** z tekstem „Użytkownik:”:
2. Dodajemy **ComboBox** o nazwie **cbUzytkownik**
3. Ustawiamy w **cbUzytkownik** właściwość **DropDownStyle** na **DropDownList**.
4. Ustawiamy w **cbUzytkownik** właściwość **DisplayMember** wpisując **Nazwa**.
5. Dodajemy **Label** z tekstem „Hasło:”
6. Dodajemy **TextBox** o nazwie **txtHaslo**.
7. Ustawiamy w **txtHaslo** właściwość **UseSystemPasswordChar** na **true**.
8. Dodajemy **Button** o nazwie **btnOk** i tekście „OK”.
9. Ustawiamy w **btnOk** właściwość **DialogResult** na **OK**.
10. Dodajemy **Button** o nazwie **btnAnuluj** i tekście „Anuluj”.
11. Ustawiamy w **btnAnuluj** właściwość **DialogResult** na **Cancel**.
12. Klikamy na formatkę **Logowanie**, żeby ją zaznaczyć i w jej właściwościach ustawiamy:
 - a. **AcceptButton** na **btnOk**.
 - b. **CancelButton** na **btnAnuluj**.
 - c. **MaximizeBox** na **False**.
 - d. **MinimizeBox** na **False**.
 - e. **SizeGripStyle** na **Hide**.

Uzyskaliśmy taki efekt:



Teraz wprowadzimy dla tej formatki kod do jej obsługi.

1. Przełączamy na widok kodu dla formatki wybierając z menu **View – Code**.
2. Zamieniamy kod klasy **Logowanie** na następujący:

```
public partial class Logowanie : Form
{
    private Uchwyty Sfera { get; set; }
    public string Login { get { return ((Uzytkownik)cbUzytkownik.SelectedItem).Login; } }

    public string Haslo { get { return txtHaslo.Text; } }
    public string Uzytkownik { get { return ((Uzytkownik)cbUzytkownik.SelectedItem).Nazwa; } }

    public Logowanie(Uchwyty sfera)
    {
        Sfera = sfera;
        InitializeComponent();
    }

    private void Logowanie_Load(object sender, EventArgs e)
    {
        cbUzytkownik.Items.AddRange(PobierzUzytkownikow());
        cbUzytkownik.SelectedIndex = 0;
    }

    private object[] PobierzUzytkownikow()
    {
        var uzytkownicy = new List<Uzytkownik>();
        using (var conn = Sfera.PodajPolaczenie())
        using (var cmd = conn.CreateCommand())
        {
            cmd.CommandText = "SELECT Id, Nazwa, Login FROM ModelDanychContainer.Uzytkownicy WHERE Ukryty=0 AND NOT Osoba_Id IS NULL;";
            conn.Open();
            using (var reader = cmd.ExecuteReader())
            {
                while (reader.Read())
                {
                    var u = new Uzytkownik();
                    u.Id = reader.GetGuid(0);
                    u.Nazwa = reader.GetString(1);
                    u.Login = reader.GetString(2);
                    uzytkownicy.Add(u);
                }
            }
        }
        return uzytkownicy.ToArray();
    }

    private bool CzyDanePoprawne()
    {
        if (string.IsNullOrEmpty(Haslo) || !Sfera.SprawdzHaslo(Login, Haslo))
        {
            MessageBox.Show(this, "Podane hasło jest niepoprawne.", this.Text);
            return false;
        }
        return true;
    }

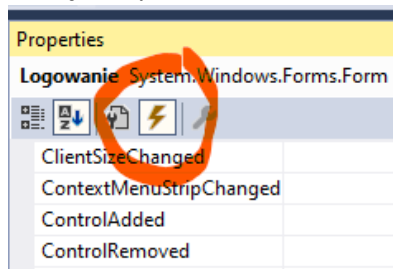
    private void Logowanie_FormClosing(object sender, FormClosingEventArgs e)
    {
        e.Cancel = DialogResult == DialogResult.OK && !CzyDanePoprawne();
    }
}
```

5. Na początku pliku dodajemy jeszcze:

```
using InSERT.Moria.Sfera;
using InSERT.Moria.ModelDanych;
```

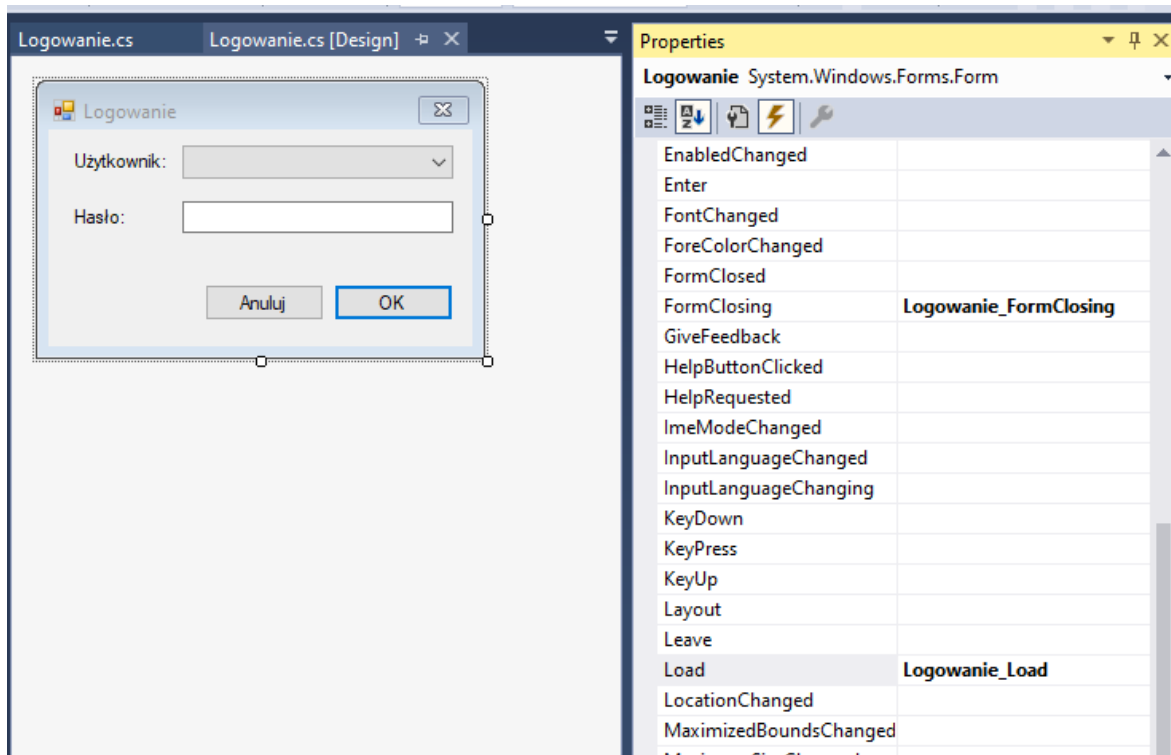
6. Przełączamy się ponownie na okno projektowania formatki **Logowanie** za pomocą menu **View – Designer**.

7. Przełączamy okno właściwości na widok zdarzeń klikając ikonkę z błyskawicą



8. Klikamy w puste pole obok **FormClosing** i z listy wybieramy **Logowanie_FormClosing**.
9. Klikamy w puste pole obok **Load** i z listy wybieramy **Logowanie_Load**.

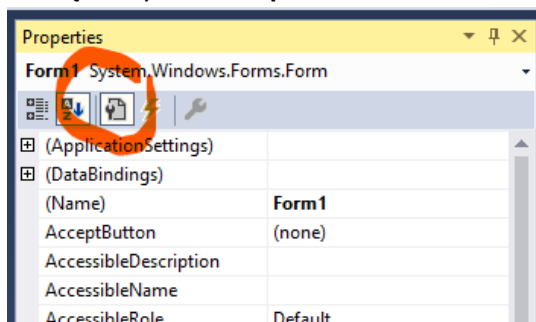
Uzyskaliśmy taki efekt



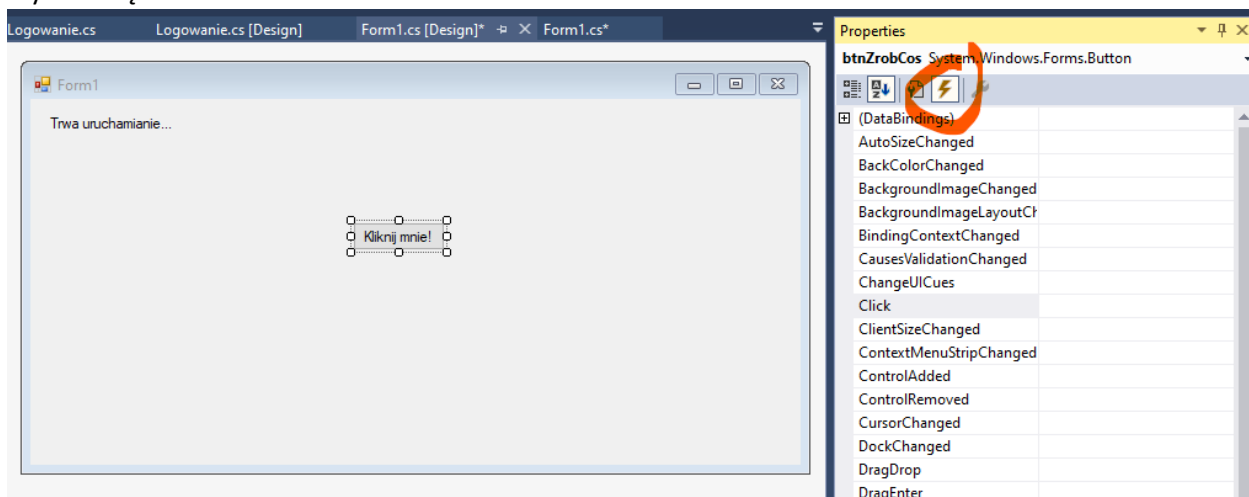
Teraz czas na wykorzystanie okna logowania w głównym oknie aplikacji

KROK 4. REALIZACJA LOGOWANIA I WYKORZYSTANIE SFERY W GŁÓWNYM OKNIE APLIKACJI

1. Przełączamy się na okno **Form1.cs [Design]** np. klikając na **Form1.cs** w oknie **Solution Explorer**.
2. Przełączamy okno **Properties** na widok właściwości:



3. Korzystając z okna **Toolbox** dodajemy na tę formatkę kontrolkę **Label**, który nazywamy **lblInformacja**.
4. W **lblInformacja** ustawiamy **Text** na „Trwa uruchamianie...”.
5. Korzystając z okna **Toolbox** dodajemy na tę formatkę **Button**, który nazywamy **btnZrobCos**.
6. We właściwościach **btnZrobCos** ustawiamy **Text** na „Kliknij mnie!”.
7. We właściwościach **btnZrobCos** właściwość **Visible** ustawiamy na **False**.
8. Mając zaznaczony przycisk **btnZrobCos** przełączamy okno właściwości na widok zdarzeń klikając ikonkę z błyskawicą



9. Dwuklikamy w pustym polu obok zdarzenia **Click** dodając nową metodę **btnZrobCos_Click** obsługującą to zdarzenie. Otwiera się nam widok kodu dla tej formatki.
10. Metodę **btnZrobCos_Click** wypełniamy następująco:

```
private void btnZrobCos_Click(object sender, EventArgs e)
{
    IPodmioty podmioty = Program.Sfera.PodajObiektTypu<IPodmioty>();

    IMojaFirma mojaFirma = podmioty.ZnajdzMojaFirme();
    MessageBox.Show(this,
        "Nazwa: " + mojaFirma.Dane.NazwaSkrocona + "\n" +
        "Adres: " + mojaFirma.Dane.AdresPodstawowy.LiniaCalosc + "\n" +
        "NIP: " + mojaFirma.Dane.NIPsformatowany,
        "Dane firmy");
}
```

1. Na początku pliku dodajemy:

```
using INSERT.Moria.Klienci;
```

11. Przełączamy się ponownie na okno projektowania formatki **Form1** za pomocą menu **View – Designer**.

12. Dwuklikamy w pustym polu obok zdarzenia **Shown** dodając nową metodę **Form1_Shown** obsługującą to zdarzenie. Otwiera się nam widok kodu.
13. Metodę **Form1_Shown** wypełniamy następująco:

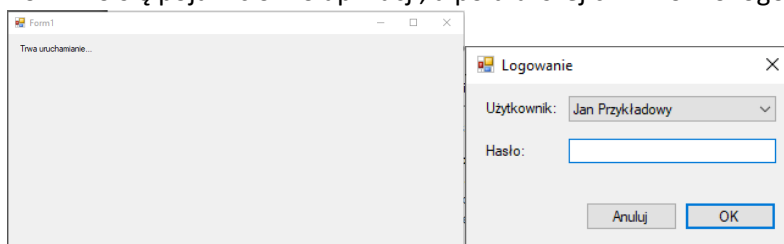
```
private void Form1_Shown(object sender, EventArgs e)
{
    Refresh();

    Logowanie logowanie = new Logowanie(Program.Sfera);
    var result = logowanie.ShowDialog();
    if (result == DialogResult.OK)
    {
        Program.Sfera.ZalogujOperatora(logowanie.Login, logowanie.Haslo);
        lblInformacja.Text = "Zalogowany użytkownik: " + logowanie.Uzytkownik;
        btnZrobCos.Visible = true;
    }
    else
    {
        Close();
    }
}
```

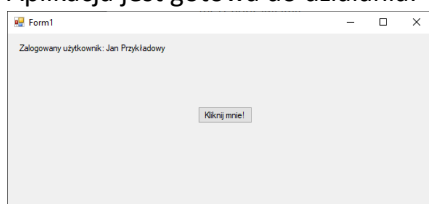
KROK 5. BUDOWANIE I URUCHAMIANIE (SPRYTNE)

Ponieważ nie wszystkie zależności między bibliotekami (assembly) są jawne nie wystarczy dodanie referencji do projektu, żeby uzyskać komplet plików w katalogu wynikowym wymagany do uruchomienia. Pomoc do Sfery opisuje jak to zrobić, ale możemy posłużyć się prostym i szybkim obejściem tego problemu budując rozwiązanie bezpośrednio do folderu nexo SDK\Bin, gdzie znajdują się wszystkie wymagane pliki.

1. Wybieramy z menu opcję **Project – SferaWinFormsApp Properties...**
2. W oknie z właściwościami projektu wberamy zakładkę **Build**.
3. W pole **Output Path** wpisujemy ścieżkę naszego folderu nexo SDK\Bin, np. **c:\nexoSDK_29.0.0.3388\Bin**
4. Budujemy rozwiązanie wybierając z menu opcję **Build – Build Solution**. Powinno się udać bez błędów. Możemy przystąpić do uruchomienia.
5. Jeśli w **Kroku 3** podaliśmy poprawne dane dostępu do serwera i bazy to już jesteśmy gotowi, a jeśli nie, to je teraz poprawiamy.
Alternatywą jest uruchomienie za pomocą InsLaunchera, ale jest to scenariusz, który będziemy mogli zrealizować dopiero po instalacji rozwiązania.
6. Uruchamiamy aplikację na jeden z poniższych sposobów:
 - a. Wybieramy z menu **Debug – Start Without Debugging** lub
 - b. Uruchamiamy **SferaWinFormsApp.exe** z folderu nexo SDK\Bin, np. **c:\nexoSDK_29.0.0.3388\Bin**.
7. Powinno się pojawić okno aplikacji, a po dłuższej chwili okno logowania:



8. Logujemy się na pracownika **Jan Przykładowy** wybierając go z listy i wpisując hasło: **robocze** i wciskając **OK**
9. Aplikacja jest gotowa do działania:



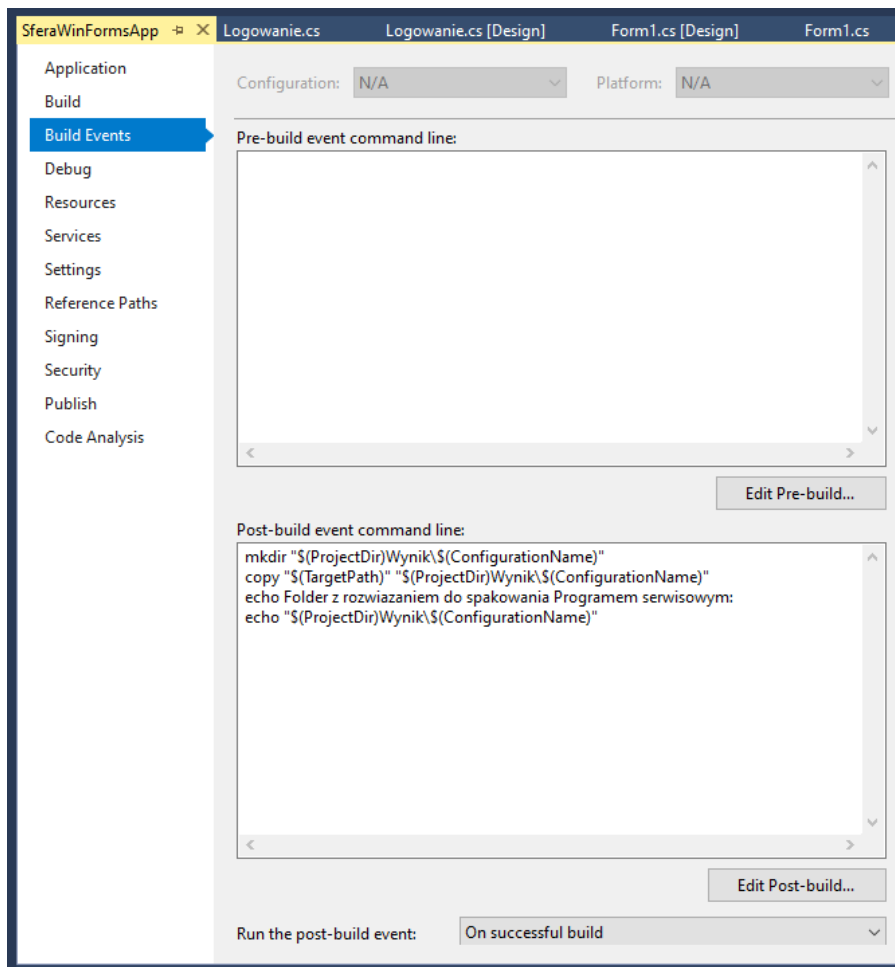
10. Klikając przycisk „Kliknij mnie!” uruchamimy metodę wykorzystującą Sferę do wyświetlenia informacji o podmiocie, do którego się zalogowaliśmy.

KROK 6. DODANIE UŁATWIENIA DLA ZLOKALIZOWANIA I SPAKOWANIA ROZWIĄZANIA

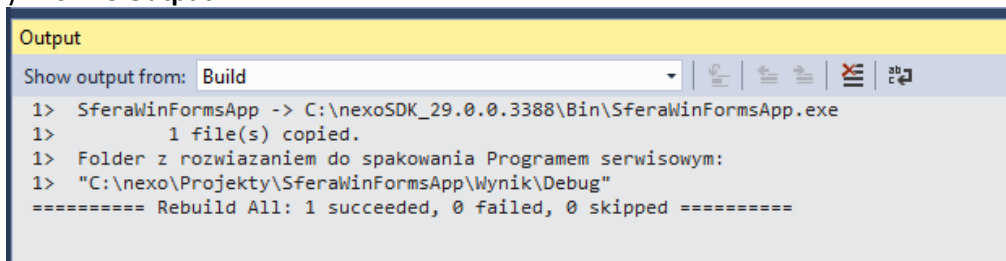
Poniższe operacje doprowadzą nas do sytuacji, z której otrzymamy rozwiązanie łatwe do spakowania i dystrybucji.

1. Wybieramy z menu opcję **Project – SferaWinFormsApp Properties...**
2. W oknie z właściwościami projektu wbieramy zakładkę **Build Events**.
3. W pole **Post-build event command line** wklejamy poniższy zestaw poleceń

```
mkdir "$(ProjectDir)Wynik\$(ConfigurationName)"
copy "$(TargetPath)" "$(ProjectDir)Wynik\$(ConfigurationName)"
echo Folder z rozwiązaniem do spakowania Programem serwisowym:
echo "$(ProjectDir)Wynik\$(ConfigurationName)"
```



Dzięki temu, po zbudowaniu mamy wyodrębniony jedyny plik do spakowania w folderze, który zostaje wydrukowany w oknie **Output**:



Koniec