

1. Programowanie usług sieciowych SOAP i konsumentów usług sieciowych

Cele ćwiczenia

- Zapoznanie z tworzeniem usług sieciowych SOAP w środowisku Visual Studio.
- Zapoznanie z opisami usług sieciowych.
- Opanowanie umiejętności tworzenia usług sieciowych i konsumentów usług sieciowych.
- Opanowanie umiejętności wykorzystania usług sieciowych w aplikacjach internetowych i Windows Forms.

Usługa sieciowa SOAP

- Usługa sieciowa dostarcza metody sieciowe, które może wywoływać aplikacja klienta.
- Usługa sieciowa nie dostarcza interfejsu użytkownika.
- Komunikacja między klientem (konsumentem) i usługą sieciową odbywa się najczęściej za pomocą bazującego na XML protokołu SOAP (ang. Simple Object Access Protocol) lub w formacie JSON (JavaScript Object Notation)– można stosować i inne formaty.
- Aktualne wersje protokołu SOAP: 1.1 i 1.2.

Tworzenie usługi sieciowej SOAP

1. Utwórz nową pustą witrynę szablonu ASP.NET Empty Web Site (ASP.NET WEB Application (.NET Framework)).
2. Do witryny dodaj usługę sieciową *Web Service (Web Service(ASMX))*.
3. Dodaj do usługi metodę sumującą dwie liczby typu double – listing 15.1.

Listing 15.1. Kod metody sieciowej dodającej dwie liczby double

```
[WebMethod(Description="Dodawanie liczb double")]
public double Dodawanie(double x, double y)
{
    return x + y;
}
```

4. Uruchom stronę z usługą zapoznaj się z dokumentacją usługi, przetestuj działanie utworzonej metody posługując się formularzem na stronie z opisem.
5. Zapoznaj się z dokumentacją usług umieszczonych pod adresami:
 - <http://argo.umg.edu.pl/www/serwis/service.asmx>,
 - <http://argo.umg.edu.pl/www/serwisSQL/service.asmx>.
6. Przetestuj wybrane metody usługi a wysyłając żądania za pomocą programu Postman lub Node-RED.

Tworzenie konsumenta usługi sieciowej SOAP

7. Dodaj do witryny referencję do usługi sieciowej <http://wekrmpc15.am.gdynia.pl/www/serwis/service.asmx>.
8. Dodaj do witryny nową stronę sumującą dwie liczby za pomocą usługi sieciowej – listing 15.2.

Listing 15.2. Kod tworzący konsumenta usługi sieciowej i wykorzystujący metodę sieciową `dodawanie()` do sumowania liczb typu double

```
try
{
```

```
ServiceReference1.ServiceSoapClient klient = new ServiceReference1.ServiceSoapClient("
ServiceSoap12");
```

```

        double x, y, suma;
        x = Convert.ToDouble(TextBox1.Text);
        y = Convert.ToDouble(TextBox2.Text);
        suma = klient.dodawanie(x, y);
        TextBox3.Text = suma.ToString();
        Label1.Text = "";
    }
    catch (Exception ex)
    {
        Label1.Text = ex.Message;
    }
}

```

9. Dodaj do witryny nową stronę umieszczającą w kontrolce GridView1 zawartość tabeli *studenci* zwracanej przez sieciową metodę *studenci()* – listing 15.3.

Listing 15.3. Kod tworzący konsumenta usługi sieciowej i wykorzystujący metodę sieciową *studenci()* do pobrania zawartości tabeli *studenci* z bazy danych

```

try
{
    ServiceReference1.ServiceSoapClient klient = new ServiceReference1.ServiceSoapClient("ServiceSoap12");
    System.Data.DataTable tabela;
    tabela = klient.studenci();
    GridView1.DataSource = tabela;
    GridView1.DataBind();
    Label1.Text = "";
}
catch (Exception ex)
{
    Label1.Text = ex.Message;
}

```

10. Przetestuj działanie strony.
11. Dodaj do witryny referencję do usługi sieciowej
<http://wekrmpc15.am.gdynia.pl/www/serwisSQL/service.asmx>.
12. Dodaj do witryny nową stronę umieszczającą w kontrolce GridView1 zawartość tabeli *student* zwracanej przez sieciową metodę *bazaStudentow()* w obiekcie klasy *DataSet1* – listing 15.4.

Listing 15.4. Kod tworzący konsumenta usługi sieciowej i wykorzystujący metodę sieciową *bazaStudentow()* do pobrania zawartości tabeli *student* z bazy danych *studenci*

```

try
{
    ServiceReference2.ServiceSoapClient klient = new ServiceReference2.ServiceSoapClient("ServiceSoap121");
    System.Data.DataTable tabela;
    ServiceReference2.DataSet1 baza;
    baza = klient.bazaStudentow();
    tabela = baza.Tables["student"];
    GridView1.DataSource = tabela;
    GridView1.DataBind();
    Label1.Text = "";
}
catch (Exception ex)
{
    Label1.Text = ex.Message;
}

```

}

13. Przetestuj działanie utworzonych stron.

Tworzenie aplikacji *Windows Forms* korzystających z usług sieciowych

1. Utwórz nowy projekt Visual C# szablonu *Aplikacja Windows Forms (.NET Framework) Application Windows Forms (.NET Framework)*.
2. Dodaj do aplikacji referencję do usługi sieciowej <http://wekrmpc15.am.gdynia.pl/www/serwis/service.asmx>.
3. Utwórz klienta usługi.
4. Na formularzu umieść kontrolki `button1`, `button2`, `dataGridView1`, `label1`.
5. Dodaj kod obsługi kliknięcia przycisków:
 - kliknięcie przycisku `button1` powoduje załadowanie danych do kontrolki `dataGridView1`,
 - kliknięcie przycisku `button2` powoduje zaktualizowanie bazy danych.

Listing 15.5. Kod obsługi kliknięcia przycisków, kliknięcie przycisku `button1` powoduje załadowanie danych do kontrolki `dataGridView1`, kliknięcie przycisku `button2` powoduje zaktualizowanie bazy danych

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        dataGridView1.DataSource = klient.studenci();
        label1.Text = "Załadowano nowe dane";
    }
    catch (Exception ex)
    {
        label1.Text = ex.Message.ToString();
    }
}
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        int wiersze = klient.aktualizacja((DataTable)dataGridView1.DataSource);
        label1.Text = "Liczba zaktualizowanych wierszy = " + wiersze.ToString();
    }
    catch (Exception ex)
    {
        label1.Text = ex.Message.ToString();
    }
}
```

6. Przetestuj działanie aplikacji.
7. Utwórz nową aplikację *Windows Forms* działającą analogicznie jak poprzednia, ale korzystającą z ulgi <http://wekrmpc15.am.gdynia.pl/www/serwisSQL/service.asmx> – listing 15.6.

Jeśli w programie wystąpi błąd związany z przekroczeniem maksymalnego rozmiaru przesyłania danych, należy dokonać wpisu do pliku konfiguracyjnego aplikacji *App.config*. Wskazówka na listingu 15.8.

Listing 15.6. Kod obsługi kliknięcia przycisków, kliknięcie przycisku `button1` powoduje załadowanie danych do kontrolki `dataGridView1`, kliknięcie przycisku `button2` powoduje zaktualizowanie bazy danych

`ServiceReference1.DataSet1 baza = new ServiceReference1.DataSet1();`

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
```

```

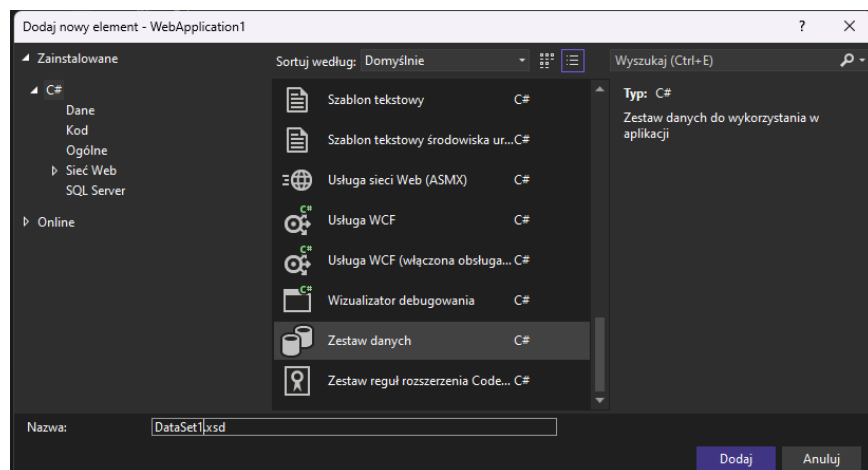
        baza = klient.bazaStudentow();
        dataGridView1.DataSource = baza.student;
        label11.Text = "Załadowano nowe dane";
    }
    catch (Exception ex)
    {
        label11.Text = ex.Message.ToString();
    }
}
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        int wiersze = klient.bazaStudentowAktualizacja(baza);
        label11.Text = "Liczba zaktualizowanych wierszy = " + wiersze.ToString();
    }
    catch (Exception ex)
    {
        label11.Text = ex.Message.ToString();
    }
}
}

```

8. Przetestuj działanie aplikacji.

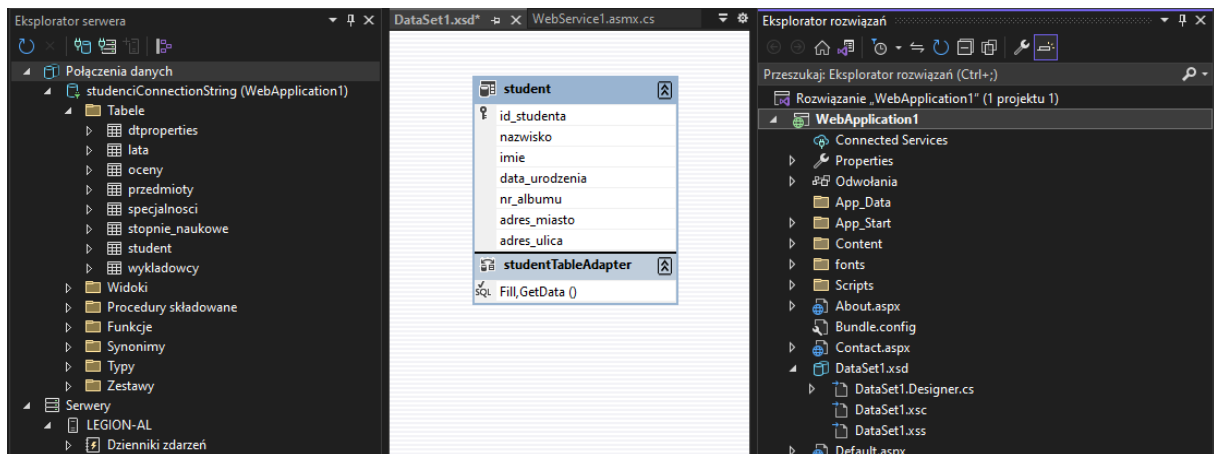
Tworzenie usługi sieciowej

Utwórz usługę sieciową zawierającą metody sieciowe, z których korzystałeś w poprzednim punkcie – listing 15.7. W kodzie korzysta się z zestawu danych DataSet1 , który należy dodać do projektu , rys. 1.



Rys. 1. Dodanie do projektu zestawu danych DataSet

Tworząc DataSet1 należy skonfigurować połączenie z serwerem bazy danych MS SQL Server. Host: *argo.am.gdynia.pl*, baza danych: *studenci*. Na rys. 2 konfiguracja zestawu danych.



Rys. 2. Konfiguracja zestawu danych DataSet1

Listing 15.7. Kod metod sieciowych wykorzystywanych w poprzednim zadaniu.

```
[WebMethod(Description="Zwraca obiekt DataSet1 reprezentujący strukturę bazy danych
studenti (MS SQLServer).<br />Tylko tabela student jest wypełniona danymi")]
```

```
public DataSet1 bazaStudentow()
{
    DataSet1 baza = new DataSet1();
```

```
    DataSet1TableAdapters.studentTableAdapter adapter_studenta = new DataSet1TableAdapters
        .studentTableAdapter();
    adapter_studenta.Fill(baza.student);
    return baza;
}
```

```
[WebMethod(Description = "Pobiera obiekt DataSet1 z wypełnioną danymi tabelą student i
aktualizuje tabelę na serwerze - baza danych studenti MS SQLServer.<br />Zwraca liczbę
zaktualizowanych wierszy")]
```

```
public int bazaStudentowAktualizacja(DataSet1 baza)
{
```

```
    DataSet1TableAdapters.studentTableAdapter adapter_studenta = new DataSet1TableAdapters
        .studentTableAdapter();
```

```
    int wiersze = adapter_studenta.Update(baza);
    return wiersze;
}
```

9. Przetestuj działanie utworzonych metod usługi sieciowej.

Listing 15.8. Kod poprawki, którą trzeba dokonać w pliku konfiguracyjnym, gdy rozmiar odpowiedzi przekroczy domyślny rozmiar maksymalny

```
<system.serviceModel>
  <bindings>
  <customBinding>
    <binding name="ServiceSoap12">
      <textMessageEncoding messageVersion="Soap12" />
      <httpTransport maxReceivedMessageSize="262144" maxBufferSize="262144" />
    </binding>
```