МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГИЙ УПРАВЛІННЯ

Лабораторна робота № 1

з курсу «Стек технологій .Net ч.2»

Виконали:

студенти групи КН 36-б

Гарбозюк М.М.

Момот М.І.

Перевірила:

Літвінова Ю.С.

ХАРКІВ 2019

Лабораторная работа № 1

WEB-СЕРВИСЫ

**Цель работы**

Познакомиться с технологией создания и использования Web-

сервисов.

**Порядок выполнения работы**

1. Создать Web-сервис на основе примера в п. 2.4, который возвра-

щал бы данные, извлеченные из БД (см. лаб. работу № 6).

1. Модифицировать Web-приложение, разработанное в предыду-

щей работе, таким образом, чтобы оно получало данные из сервиса,

созданного в предыдущем пункте.

**Ход работы**

Для создания веб-сервиса создадим новы й проект в Visual Studio: ASP.NET Web Service Application.

Изменим код созданного класса сервиса, создав в нем три метода: для чтения, добавления и удаления данных с БД с последующим формированием результирующего DataSet.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

using System.Data;

using System.Data.SqlClient;

namespace lab7

{

/// <summary>

/// Summary description for nw

/// </summary>

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

// [System.Web.Script.Services.ScriptService]

public class nw : System.Web.Services.WebService

{

[WebMethod]

public DataSet GetInfo(string conn)

{

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter("Select \* from Product", conn);

DataSet productsDataSet = new DataSet();

sqlDataAdapter.Fill(productsDataSet, "Products");

return productsDataSet;

}

[WebMethod]

public DataSet Add(string conn, string name, string developer, decimal price)

{

using (SqlConnection connection = new SqlConnection(conn))

{

connection.Open();

using (SqlCommand cmd = new SqlCommand("insert into Product values(@name, @price, @dev)", connection))

{

cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;

cmd.Parameters.Add("@dev", SqlDbType.NVarChar).Value = developer;

cmd.Parameters.Add("@price", SqlDbType.Decimal).Value = price;

cmd.ExecuteNonQuery();

}

}

return GetInfo(conn);

}

[WebMethod]

public DataSet Delete(string conn, int id)

{

using (SqlConnection connection = new SqlConnection(conn))

{

connection.Open();

using (SqlCommand cmd = new SqlCommand("delete from Product where id = @id", connection))

{

cmd.Parameters.Add("@id", SqlDbType.Int).Value = id;

cmd.ExecuteNonQuery();

}

}

return GetInfo(conn);

}

}

}

Листинг 1 – Код сервиса

Протестируем работу сервиса в браузере. Для этого запустим проект и выполним один из описанных выше веб-методов.

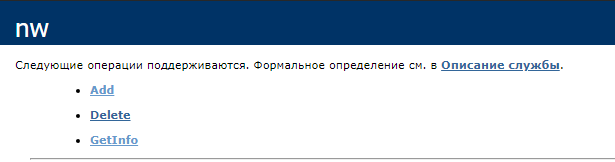


Рисунок 1 – Страница веб-службы

This XML file does not appear to have any style information associated with it. The document tree is shown below.

<DataSet xmlns="http://tempuri.org/">

<xs:schema xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" id="NewDataSet">

<xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">

<xs:complexType>

<xs:choice minOccurs="0" maxOccurs="unbounded">

<xs:element name="Products">

<xs:complexType>

<xs:sequence>

<xs:element name="id" type="xs:int" minOccurs="0"/>

<xs:element name="name" type="xs:string" minOccurs="0"/>

<xs:element name="price" type="xs:decimal" minOccurs="0"/>

<xs:element name="developer" type="xs:string" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:choice>

</xs:complexType>

</xs:element>

</xs:schema>

<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">

<NewDataSet xmlns="">

<Products diffgr:id="Products1" msdata:rowOrder="0">

<id>1</id>

<name>Комп'ютер</name>

<price>5555.00</price>

<developer>Toshiba</developer>

</Products>

<Products diffgr:id="Products2" msdata:rowOrder="1">

<id>2</id>

<name>Смартфон</name>

<price>8000.00</price>

<developer>Samsung</developer>

</Products>

<Products diffgr:id="Products3" msdata:rowOrder="2">

<id>3</id>

<name>Стройматеріали</name>

<price>200.00</price>

<developer>Kingston</developer>

</Products>

<Products diffgr:id="Products4" msdata:rowOrder="3">

<id>4</id>

<name>Кофеварка</name>

<price>3500.00</price>

<developer>Royality</developer>

</Products>

<Products diffgr:id="Products5" msdata:rowOrder="4">

<id>5</id>

<name>Холодильик</name>

<price>3900.00</price>

<developer>Wacoom</developer>

</Products>

<Products diffgr:id="Products6" msdata:rowOrder="5">

<id>6</id>

<name>Телепортатор</name>

<price>100000.00</price>

<developer>CERN</developer>

</Products>

<Products diffgr:id="Products7" msdata:rowOrder="6">

<id>1008</id>

<name>brand new</name>

<price>11.00</price>

<developer>china</developer>

</Products>

<Products diffgr:id="Products8" msdata:rowOrder="7">

<id>1015</id>

<name>ручка</name>

<price>12.00</price>

<developer>україна</developer>

</Products>

<Products diffgr:id="Products9" msdata:rowOrder="8">

<id>1017</id>

<name>Діма</name>

<price>1.00</price>

<developer>мама</developer>

</Products>

<Products diffgr:id="Products10" msdata:rowOrder="9">

<id>1018</id>

<name>Картина</name>

<price>100000.00</price>

<developer>Пикасо</developer>

</Products>

</NewDataSet>

</diffgr:diffgram>

</DataSet>

Листинг 2 – полученный SOAP файл

Далее следует создать ссылку на сервис в клиентском приложении. В качестве приложения-клиента используем веб-приложение, разработанное в результате предыдущей лабораторной работы.

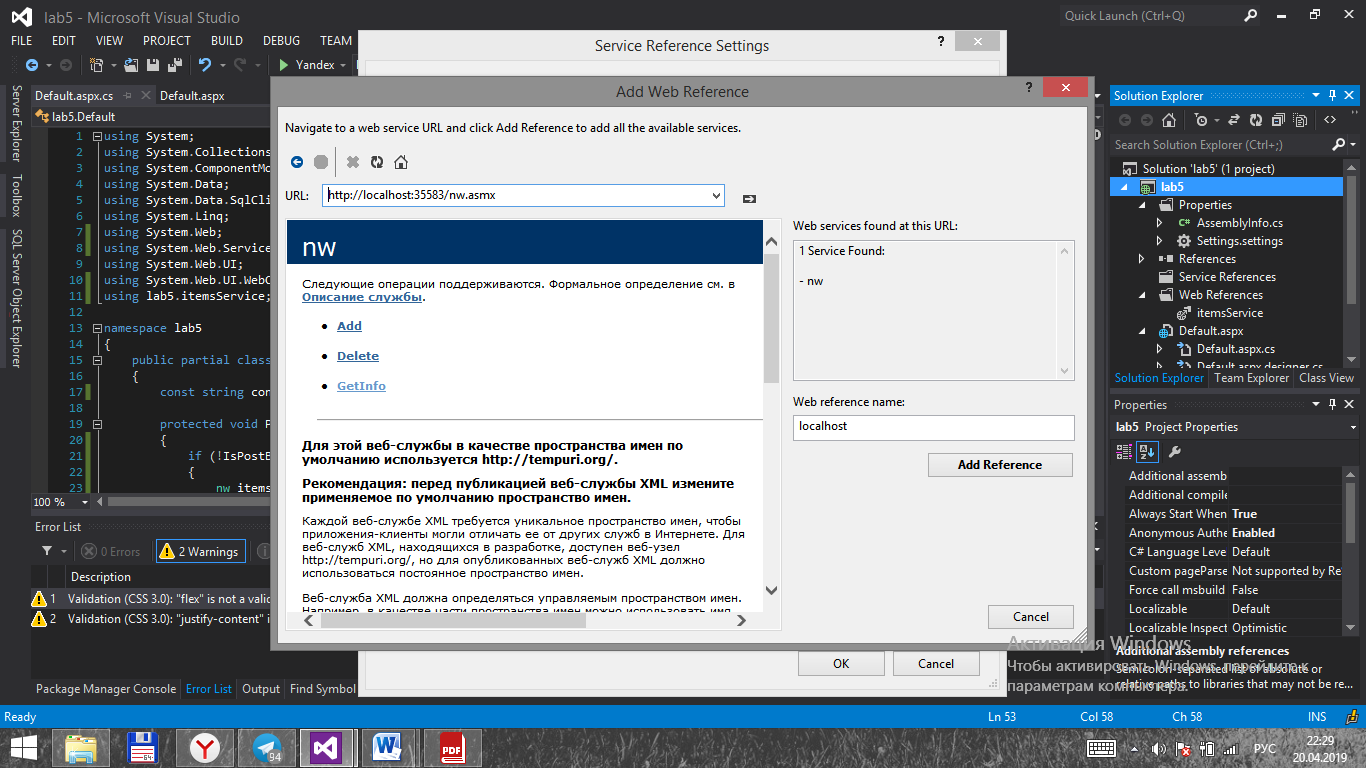


Рисунок 2 – Создание ссылки на веб-сервис

Для этого воспользуемся URL ссылкой веб-сервиса. Во время создания ссылки, сервис должен быть запущен на другом экземпляре Visual Studio.

После создания ссылки, можно создать экземпляр сервиса в программе и модифицировать методы в веб-приложении используя методы веб-сервиса.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.Services.Description;

using System.Web.UI;

using System.Web.UI.WebControls;

using lab5.itemsService;

namespace lab5

{

public partial class Default : System.Web.UI.Page

{

const string conn = @"Data Source=ADMIN;Initial Catalog=lab6;Integrated Security=True";

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

nw items = new nw();

GridView2.DataSource = items.GetInfo(conn);

GridView2.DataBind();

}

}

protected void GridView2\_OnRowCommand(object sender, GridViewCommandEventArgs e)

{

if (e.CommandName != "Delete") return;

int id = Convert.ToInt32(e.CommandArgument) ;

nw items = new nw();

GridView2.DataSource = items.Delete(conn, id);

GridView2.DataBind();

}

protected void Add(object sender, EventArgs e)

{

if (TextBox1.Text == string.Empty || TextBox2.Text == string.Empty || TextBox3.Text == string.Empty)

{

ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "alertMessage", "alert('please complete all the fields')", true);

return;

}

nw items = new nw();

string name = TextBox1.Text;

string developer = TextBox2.Text;

decimal price = decimal.Parse(TextBox3.Text);

GridView2.DataSource = items.Add(conn, name, developer, price);

GridView2.DataBind();

TextBox1.Text = String.Empty;

TextBox2.Text = String.Empty;

TextBox3.Text = String.Empty;

}

protected void GridView2\_RowDeleting(object sender, GridViewDeleteEventArgs e)

{

nw items = new nw();

GridView2.DataSource = items.GetInfo(conn);

GridView2.DataBind();

}

}

}

Листинг 3 – Codebehind веб-страницы



Рисунок 3 – Результат работы веб-приложения

**Выводы**

На данной лабораторной работе были изучены основные особенности использования веб-сервисов ASP.NET и интеграция их в клиентские приложения. Было усовершенствованно веб-приложение, разработанное на прошлой лабораторной работе, путем интеграции в него разработанного веб-сервиса. Разработанный веб-сервис реализует методы чтения, добавления и удаления данных в БД.