

## Практическое задание № 6. Веб-сервисы (web-service)

### Теория

Материал составлен на основе ресурса: Горин С.В., Крищенко В.А. Поддержка разработки распределенных приложений в Microsoft .NET Framework. Учебный курс. [Электронный ресурс]. URL: <http://window.edu.ru/resource/701/41701/files/ds-dotnet.pdf> (дата обращения: 15.01.2014).

Веб-службой или веб-сервисом (web-service, WS) называется программная компонента, предоставляющая сервис удаленного вызова на основе группы стандартов WSI (Web Services Interoperability), основными из которых являются протокол обмена сообщениями SOAP, язык описания интерфейса WSDL, HTTP как основной транспортный протокол, а также XML и схемы XML. Для описания спецификаций формата сообщений в веб-службах в настоящее время обычно используется схема XML и кодирование тела пакета SOAP Document.

Служба ASP.NET организует на основе данных стандартов сервис удаленного вызова методов объектов. ASP.NET создает и публикует WSDL описание веб-службы на основе интерфейса класса .NET, обрабатывает приходящие от клиентов сообщения SOAP, вызывает методы объекта с извлеченными из сообщений SOAP параметрами, а затем передает клиенту сообщение с результатом выполнения метода (рис. 1). Таким образом, полное название промежуточной среды – веб-службы ASP.NET.

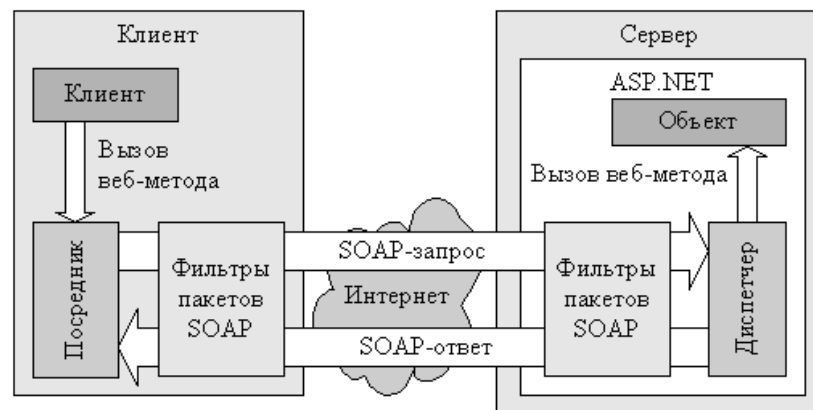


Рис. 1 Архитектура веб-служб ASP.NET

Хотя с точки зрения прикладного разработчика использование веб-службы представляется вариантом удаленного вызова через интернет, в основе веб-служб лежит расширяемая технология обмена сообщениями SOAP, независимая от транспортного уровня. В частности, существуют стандарты, описывающие маршрутизацию сообщений SOAP, что придает веб-службам признаки среды обмена сообщениями. Благодаря гибкой технологии возможно расширение промежуточной среды веб-служб как по мере принятия новых стандартов, так и при необходимости в нестандартной обработке пакетов SOAP.

Веб-сервисы наиболее универсальная, открытая, масштабируемая и надежная технология построения распределенных систем. Реализация механизма расширений WSE достаточно понятна и проста для использования разработчиком. После повсеместного принятия полного набора стандартов WS-\* веб-службы будут обладать широкими возможностями поддержки распределенных транзакций. Также определенным недостатком веб-служб ASP.NET можно считать отсутствие в настоящий момент поддержки локального взаимодействия, не затрагивающего стек TCP/IP.

### Функции для программной реализации веб-сервисов

Описание интерфейса веб-сервисов выполняется на языке WSDL (Web Service Description Language).

Описание функциональности веб-сервиса производится на C#, VB, C++ или другом объектно-ориентированном языке программирования.

### Алгоритм разработки веб-службы и клиента для работы с ней

Разработка веб-сервиса и клиента для работы с ним выполняется по следующему алгоритму:

1. Создание проекта для разработки веб-службы. Окно создания такого проекта в среде MS Visual Studio (.Net Framework 3.5) представлено на рис. 2.

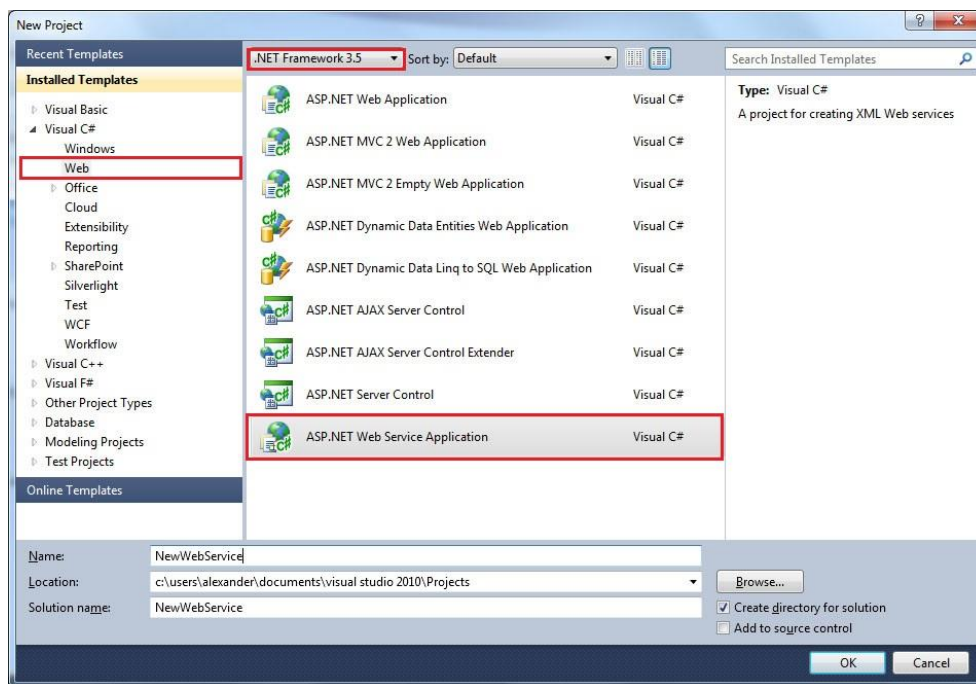


Рис. 2. Окно создания проекта для разработки веб-сервиса

2. Описание функциональности веб-сервиса.
3. Запуск веб-сервиса на выполнение. При этом описание веб-службы публикуется на ASP.Net Development Server (см. рис. 3).

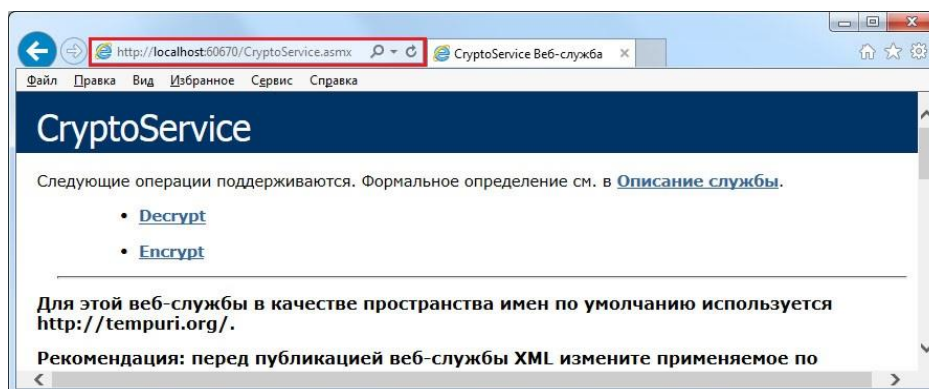


Рис. 3. Окно браузера с описанием веб-сервиса

4. Создание клиента для работы со службой (WinForms Application, Console Application, веб-сайта и др.).
5. Добавление ссылки на опубликованный веб-сервис в клиентском приложении (см. рис. 4-5).

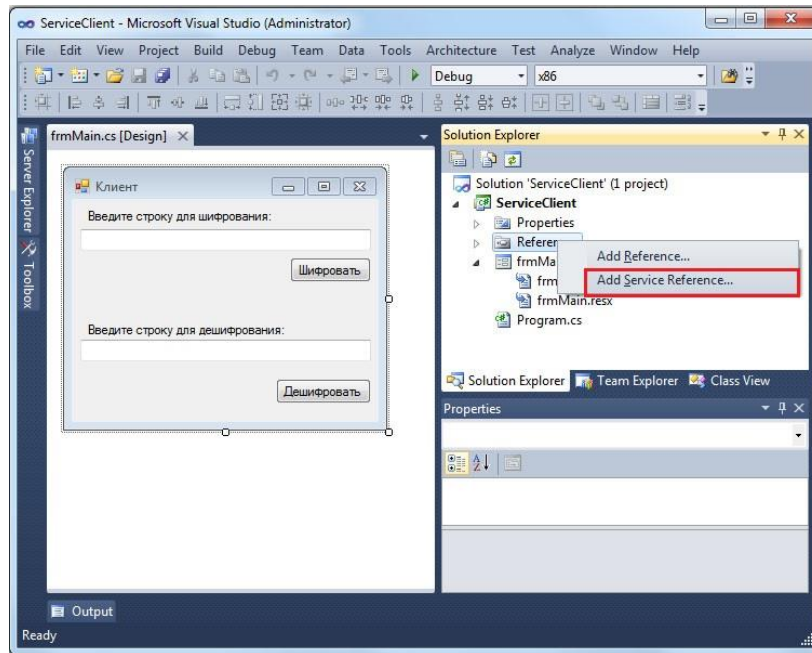


Рис. 4. Добавление ссылки на веб-сервис. Этап I

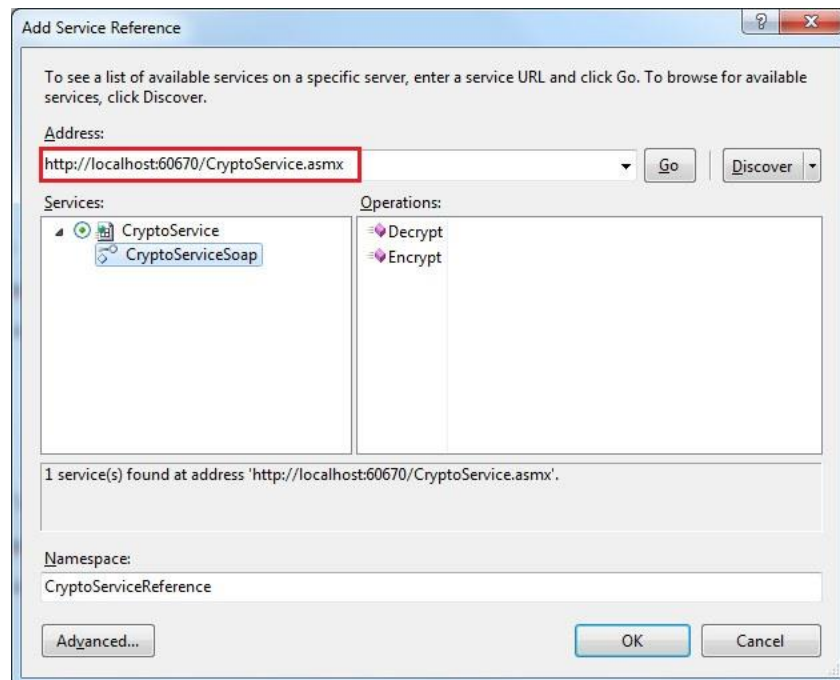


Рис. 5. Добавление ссылки на веб-сервис. Этап II

6. Написание программной реализации клиента, использующего веб-сервис.

### Разработка веб-сервиса

Разработаем веб-сервис, выполняющий шифрование/дешифрование переданной ему строки с помощью шифра Цезаря. Веб-сервис будет содержать два удаленных метода: первый используется для шифрования строки, второй – для ее дешифрования. Для демонстрации передачи параметров в веб-сервис, создадим собственный класс, описывающий шифруемую строку.

Клиентское приложение представляет собой форму, содержащую два поля ввода и две кнопки (см. рис. 6). Первое поле предназначено для ввода шифруемой строки, второе – для ввода дешифруемой строки. При нажатии на кнопку «Шифровать»

выполняется шифрование введенной пользователем строки, при нажатии на кнопку «Дешифровать» выполняется дешифрование введенной пользователем строки.

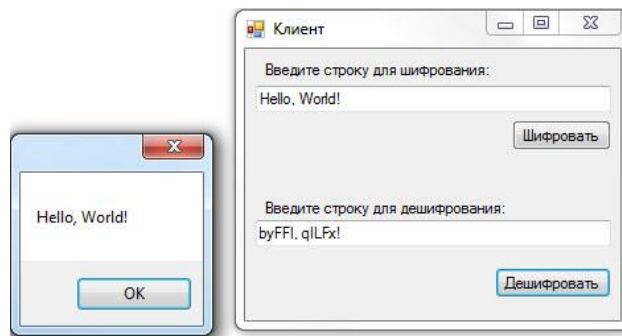


Рис. 6. Окно клиентского приложения, использующего веб-службу

Код реализации веб-сервиса с комментариями приведен ниже:

```
namespace NewWebService
{
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    public class CryptoService : System.Web.Services.WebService
    {
        private static string alphabet =
            "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"; // строка, содержащая все буквы
            английского алфавита
        private int n = alphabet.Length; // мощность алфавита
        private int k = 20; // ключ шифрования

        // метод, выполняющий шифрование строки
        [WebMethod]
        public CryptoStr Encrypt(CryptoStr str)
        {
            // создаем новый объект класса CryptoStr
            CryptoStr EncryptStr = new CryptoStr();

            // выполняем шифрование по формуле вида (y = (x + k) mod n), где x - шифруемый
            символ, k - ключ, n - мощность алфавита, y - результат шифрования, mod - операция взятия
            остатка от целочисленного деления
            for (int i = 0; i < str.Length; i++)
            {
                if (alphabet.IndexOf(str.SourceString[i]) == -1)
                    EncryptStr.SourceString += str.SourceString[i]; // если символа в
                    алфавите нет, то оставляем его без изменений (например, пробелы, знаки препинания и др.)
                else
                    EncryptStr.SourceString += alphabet[(alphabet.IndexOf(str.SourceString[i])
                    + k) % n]; // если символ есть в алфавите, то выполняем его шифрование по вышеуказанной
                    формуле
            }

            return EncryptStr;
        }

        // метод, выполняющий дешифрование строки
        [WebMethod]
        public CryptoStr Decrypt(CryptoStr str)
        {
            // создаем новый объект класса CryptoStr
            CryptoStr DecryptStr = new CryptoStr();

            // выполняем дешифрование по формуле вида (y = (x - k + n) mod n), где x -
            дешифруемый символ, k - ключ, n - мощность алфавита, y - результат дешифрования, mod -
            операция взятия остатка от целочисленного деления
            for (int i = 0; i < str.Length; i++)
                if (alphabet.IndexOf(str.SourceString[i]) == -1)
```

```
        DecryptStr.SourceString += str.SourceString[i];    // если символа в
    алфавите нет, то оставляем его без изменений (например, пробелы, знаки препинания и др.)
    else
        DecryptStr.SourceString += alphabet[(alphabet.IndexOf(str.SourceString[i])
    - k + n) % n];    // если символ есть в алфавите, то выполняем его дешифрование по
    вышеуказанной формуле

    return DecryptStr;
}
}

// класс, определяющий строку шифрования
public class CryptoStr
{
    private string _str;

    // конструктор
    public CryptoStr()
    {
        _str = "";
    }

    // строка
    public string SourceString { get { return _str; } set { _str = value; } }

    // длина строки
    public int Length { get { return _str.Length; } }
}
}
```

Код реализации клиента с комментариями приведен ниже:

```
public partial class frmMain : Form
{
    // конструктор формы
    public frmMain()
    {
        InitializeComponent();
    }

    // шифрование строки
    private void btnEncrypt_Click(object sender, EventArgs e)
    {
        // если строка не введена, то выходим из обработчика нажатия кнопки
        if (tbEncrypt.Text == null || String.IsNullOrEmpty(tbEncrypt.Text))
        {
            MessageBox.Show("Не введена строка для шифрования. Введите текст");
            return;
        }

        // создаем экземпляр класса CryptoStr и присваиваем ему значение поля tbEncrypt
        CryptoServiceReference.CryptoStr str = new CryptoServiceReference.CryptoStr();
        str.SourceString = tbEncrypt.Text;

        // вызываем метод веб-сервиса по шифрованию строки
        CryptoServiceReference.CryptoServiceSoapClient service = new
        CryptoServiceReference.CryptoServiceSoapClient();

        // выводим результат шифрования строки
        MessageBox.Show(service.Encrypt(str).SourceString);
    }

    // дешифрование строки
    private void btnDecrypt_Click(object sender, EventArgs e)
    {
        // если строка не введена, то выходим из обработчика нажатия кнопки
        if (tbDecrypt.Text == null || String.IsNullOrEmpty(tbDecrypt.Text))
        {
            MessageBox.Show("Не введена строка для дешифрования. Введите текст");
        }
    }
}
```

```
        return;
    }

    // создаем экземпляр класса CryptoStr и присваиваем ему значение поля tbDecrypt
    CryptoServiceReference.CryptoStr str = new CryptoServiceReference.CryptoStr();
    str.SourceString = tbDecrypt.Text;

    // вызываем метод веб-сервиса по дешифрованию строки
    CryptoServiceReference.CryptoServiceSoapClient service = new
CryptoServiceReference.CryptoServiceSoapClient();

    // выводим результат дешифрования строки
    MessageBox.Show(service.Decrypt(str).SourceString);
    }
}
```

### **Задание**

Разработать веб-сервис и клиентское приложение, его использующее:

1. Веб-сервис должен принимать параметры и передавать приложению результат.
2. В качестве параметров и результата должны передаваться экземпляры классов, написанных самостоятельно, т.е. не должны передаваться строки, числа или другие простейшие типы.

Функции, выполняемые веб-сервисом, студент выбирает самостоятельно по согласованию с преподавателем.