

Практическая работа №8

Цель работы: изучения способов создания Web-сервисов

Содержание работы

1. Изучить основные принципы создания Web-сервисов.
2. Создать web-сервис, который будет использовать XML документ в качестве источника данных. То есть создать Web- сервис, возвращающий обычный набор данных содержащий информацию, взятую из XML-файла.
3. Дополнить созданный web- сервис процедурой позволяющей сохранять данные в XML файле.
4. Создать клиентское приложение.
5. Оформить отчет, включающий постановку задачи, текст программ, демонстрацию работы программ.
6. Защитить лабораторную работу

Методические указания

Сервис

Для того чтобы создать самый простой Web-сервис, который по запросу пользователя будет выдавать текущую дату или комбинацию даты и времени в среде разработки Visual Studio .NET следует выполнить команду меню File | New | Project и в появившемся диалоговом окне New Project в наборе Templates выделить значок ASP.NET Web Service. В поле Name следует указать наименование создаваемого проекта. После того, как все необходимые приготовления средой разработки будут сделаны, на основном рабочем поле появятся две новые страницы. Одна будет предназначена для разработки визуального дизайна, а на второй будет располагаться код нашего сервиса. Естественно, у создаваемого сервиса не может быть внешнего вида как такового, ему нечего отображать, поэтому страницу для дизайна можно спокойно закрыть.

В состав одного проекта может входить несколько отдельных сервисов. Для нашего примера потребуется всего один Web-сервис, заготовка для которого создается средой разработки Visual Studio. NET автоматически, поэтому ничего изменять не потребуется, и мы можем сразу перейти на страницу с наименованием Service1.aspx.vb. На этой странице нужно разместим код нашего Web-сервиса:

```
Imports System.Web.Services
```

```
<WebService(Namespace := "http://tempuri.org/")> _
```

```
Public Class Service1
```

```
Inherits System.Web.Services.WebService
```

```
#Region " Web Services Designer Generated Code "
```

```
Public Sub New()
```

```
MyBase.New()
```

'This call is required by the Web Services Designer.
InitializeComponent()

'Add your own initialization code after the InitializeComponent() call

End Sub

'Required by the Web Services Designer
Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Web Services Designer
'It can be modified using the Web Services Designer.

'Do not modify it using the code editor.

<System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()

components = New System.ComponentModel.Container()

End Sub

Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)

'CODEGEN: This procedure is required by the Web Services Designer

'Do not modify it using the code editor.

If disposing Then

If Not (components Is Nothing) Then

components.Dispose()

End If

End If

MyBase.Dispose(disposing)

End Sub

#End Region

<WebMethod()> Public Function MyDate(ByVal ShowTime As Boolean
) As String

Dim MD As DateTime

If ShowTime Then

MyDate = MD.Now

Else

MyDate = MD.Today

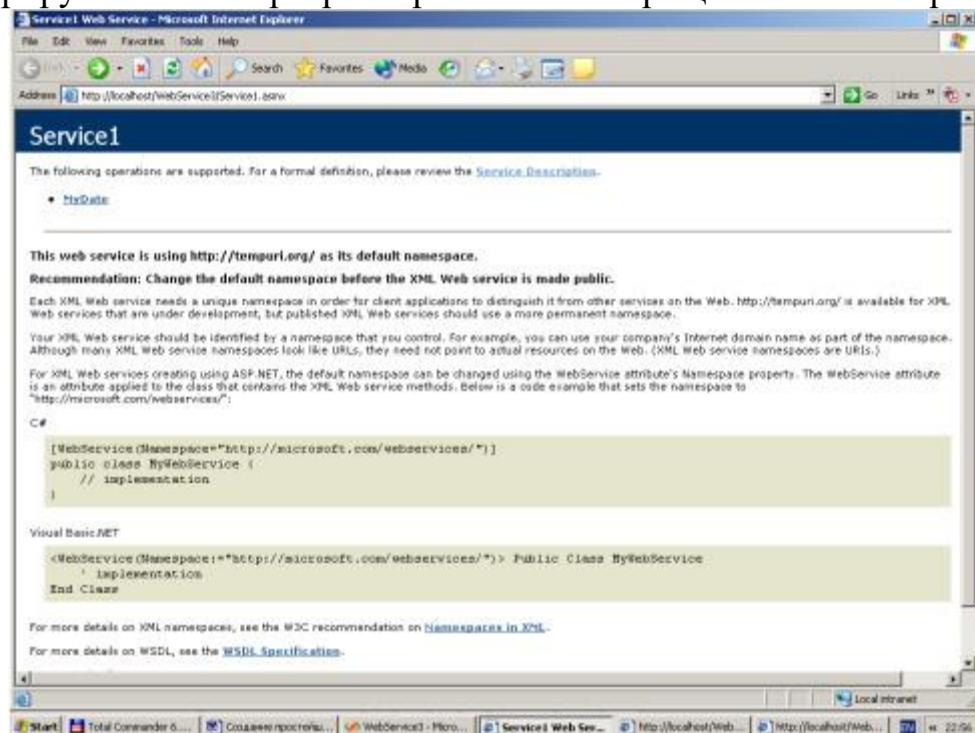
End If

End Function

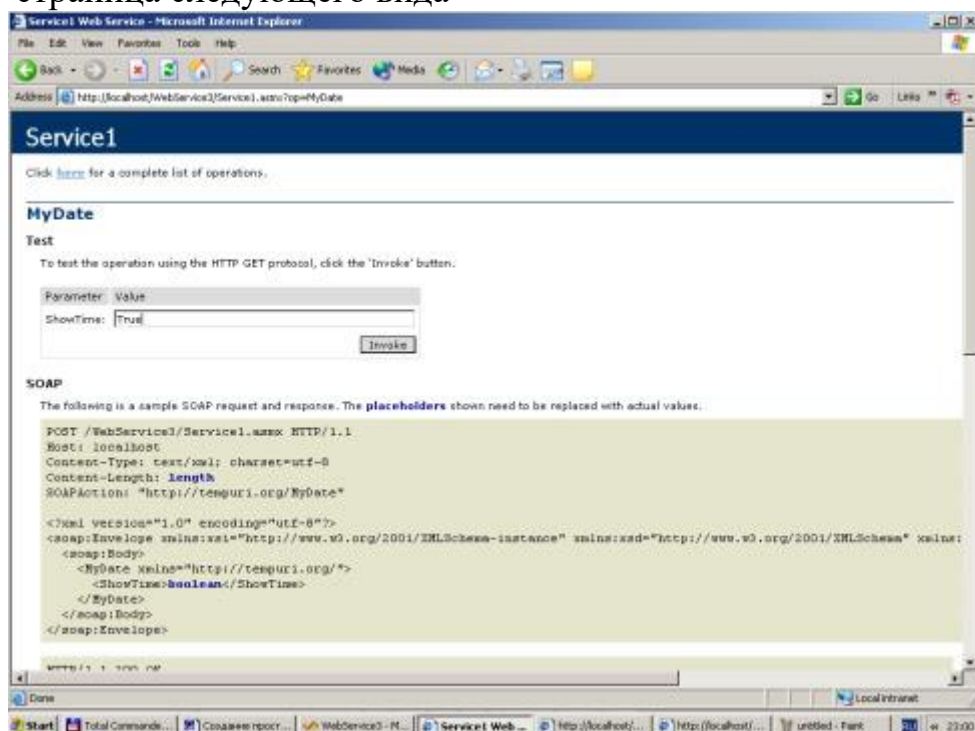
End Class

Из всего этого кода лишь малая часть написана разработчиком. Это функция MyDate, расположенная в конце листинга.

После компиляции к созданному Web сервису можно обратиться даже из браузера. На рис. приведен внешний вид Web-страницы, которая генерируется Web-сервером при попытке обращения к Web сервису.



На этой Web-странице расположена ссылка на формальное описание структуры Web-сервиса, а также перечислены все функции, поддерживаемые Web-сервисом. В нашем случае, естественно, указана лишь одна функция MyDate. Наименование функции также является гиперссылкой, нажав на которую можно перейти к Web-странице, позволяющей воспользоваться этой функцией. После щелчка мышью на ссылке в браузере будет отображена Web-страница следующего вида



На этой странице расположено поле ввода, в котором пользователь может указать значение параметра, передаваемого функции.

В результате работы функции сервиса получим:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<string xmlns="http://tempuri.org/">11.08.2006 23:03:54</string>
```

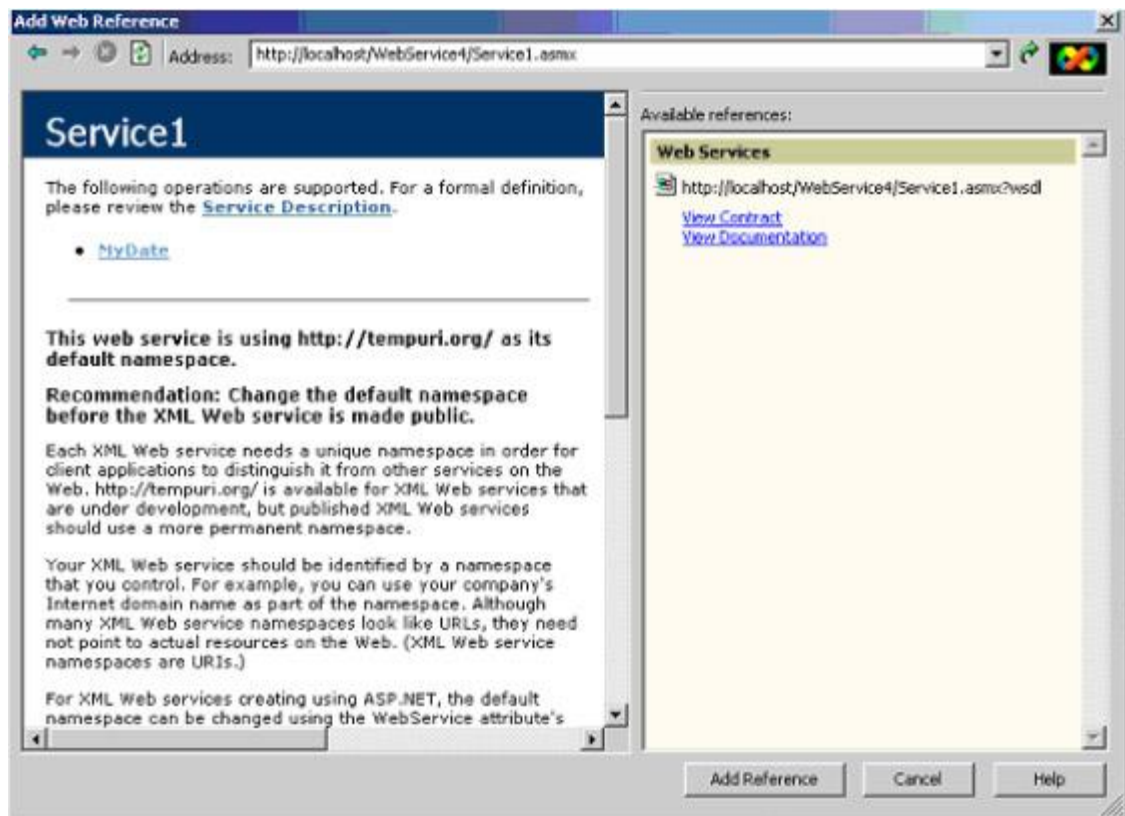
Клиентская часть

Для создания клиентского приложения нужно для начала создать новый проект. Для этого следует выполнить команду меню File | New | Project. В появившемся диалоговом окне New Project в наборе Templates выбрать шаблон Windows Application и в поле Name указать наименование создаваемого приложения.

После этого Visual Studio .NET создаст форму с именем Form1, устанавливаемым по умолчанию. В окне этой формы нам потребуется разместить один элемент Checkbox, одно текстовое поле Label, и одну кнопку Button. При помощи независимого переключателя CheckBox1 мы будем задавать параметр, передаваемый функции MyDate, входящей в состав сервиса. Текстовое поле будет содержать результат, возвращаемый этой функцией, а кнопка - запускать процесс установки связи с Web-сервисом и получения от него требуемых данных.

После этого следует установить связь с искомым Web-сервисом. Для этого надо в окне Solution Explorer выбрать наименование создаваемого приложения и правым щелчком мыши вызвать для него контекстное меню. В этом контекстном меню необходимо выполнить команду Add Web Reference, после чего будет активизировано одноименное диалоговое окно, позволяющее установить ссылки на используемые сервисы.

Для того чтобы получить ссылки на функции Web-сервиса и перенести их в проект, нужно в текстовом поле Address указать URL необходимого сервиса и загрузить ресурс, располагающийся по этому адресу. После этого в левой части искомого окна Add Web Reference будет отображено содержимое стартовой Web-страницы.



После того как искомые ссылки на функции Web-сервиса будут найдены, при помощи кнопки Add Reference их необходимо добавить к разрабатываемому проекту. После чего можно обращаться к функциям Web-сервиса в приложении (функция Button_Click).

```
Public Class Form1
```

```
    Inherits System.Windows.Forms.Form
```

```
#Region " Windows Form Designer generated code "
```

```
Public Sub New()
```

```
    MyBase.New()
```

```
    'This call is required by the Windows Form Designer.
```

```
    InitializeComponent()
```

```
    'Add any initialization after the InitializeComponent() call
```

```
End Sub
```

```
'Form overrides dispose to clean up the component list.
```

```
Protected Overrides Sub Dispose(ByVal disposing As Boolean)
```

```
    If disposing Then
```

```
        If Not (components Is Nothing) Then
```

```
            components.Dispose()
```

```
        End If
```

```
End If
MyBase.Dispose(disposing)
End Sub
```

'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

Friend WithEvents CheckBox1 As System.Windows.Forms.CheckBox

Friend WithEvents Label1 As System.Windows.Forms.Label

Friend WithEvents Button1 As System.Windows.Forms.Button

<System.Diagnostics.DebuggerStepThrough()> Private Sub

InitializeComponent()

Me.CheckBox1 = New System.Windows.Forms.CheckBox()

Me.Label1 = New System.Windows.Forms.Label()

Me.Button1 = New System.Windows.Forms.Button()

Me.SuspendLayout()

,

'CheckBox1

,

Me.CheckBox1.Location = New System.Drawing.Point(16, 16)

Me.CheckBox1.Name = "CheckBox1"

Me.CheckBox1.Size = New System.Drawing.Size(168, 24)

Me.CheckBox1.TabIndex = 0

Me.CheckBox1.Text = "Отображать время"

,

'Label1

,

Me.Label1.Location = New System.Drawing.Point(16, 64)

Me.Label1.Name = "Label1"

Me.Label1.Size = New System.Drawing.Size(264, 23)

Me.Label1.TabIndex = 1

,

'Button1

,

Me.Button1.Location = New System.Drawing.Point(16, 128)

Me.Button1.Name = "Button1"

Me.Button1.Size = New System.Drawing.Size(256, 32)

Me.Button1.TabIndex = 2

Me.Button1.Text = "Запуск Web-сервиса"

,

'Form1

```

        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.ClientSize = New System.Drawing.Size(292, 273)
        Me.Controls.AddRange(New System.Windows.Forms.Control()
{Me.Button1, Me.Label1, Me.CheckBox1})
        Me.Name = "Form1"
        Me.Text = "Form1"
        Me.ResumeLayout(False)

End Sub

#End Region

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim t1 = New WindowsApplication1.localhost.Service1()
    Label1.Text = t1.MyDate(CheckBox1.Checked)
End Sub
End Class

```

Листинг 13.1.

После запуска приложение будет выглядеть следующим образом:



XML-документ в качестве источника данных

Создадим Web-сервис, который будет включать единственную функцию, возвращающую обычный набор данных (DataSet), содержащий информацию, взятую из XML-файла.

Для этого создадим новый проект Web-сервиса. На его страницу отбуксируем один компонент DataSet с вкладки Data. При этом будет активировано диалоговое окно AddDataset, в котором разработчику предлагается указать тип создаваемого набора данных. Так как в нашем случае XML-файл может содержать данные любой структуры, то для набора данных нельзя заранее указать структуру. Следовательно, в этом диалоговом окне разработчик должен выбрать переключатель Untyped dataset.

Код Web-сервиса приведен ниже.

```
Imports System.Web.Services
```

<WebService(Namespace := "http://tempuri.org/")7> _

Public Class Service1

Inherits System.Web.Services.WebService

#Region " Web Services Designer Generated Code "

Public Sub New()

MyBase.New()

'This call is required by the Web Services Designer.

InitializeComponent()

'Add your own initialization code after the InitializeComponent() call

End Sub

'Required by the Web Services Designer

Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Web Services Designer

'It can be modified using the Web Services Designer.

'Do not modify it using the code editor.

Friend WithEvents DataSet1 As System.Data.DataSet

<System.Diagnostics.DebuggerStepThrough()>

Private

Sub

InitializeComponent()

Me.DataSet1 = New System.Data.DataSet()

CType(Me.DataSet1,

System.ComponentModel.ISupportInitialize).BeginInit()

,

'DataSet1

,

Me.DataSet1.DataSetName = "NewDataSet"

Me.DataSet1.Locale = New System.Globalization.CultureInfo("ru-RU")

CType(Me.DataSet1,

System.ComponentModel.ISupportInitialize).EndInit()

End Sub

Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)

'CODEGEN: This procedure is required by the Web Services Designer

'Do not modify it using the code editor.

If disposing Then

If Not (components Is Nothing) Then

components.Dispose()


```

        End If
    End If
    MyBase.Dispose(disposing)
End Sub

```

```
#End Region
```

```

    <WebMethod()> Public Function XMLData(ByVal file As String) As
DataSet
    Dim FS As IO.FileStream
    Dim Reader As IO.StreamReader
    FS = New IO.FileStream(Server.MapPath(file), IO.FileMode.Open,
IO.FileAccess.Read)
    Reader = New IO.StreamReader(FS)
    DataSet1.ReadXml(Reader)
    FS.Close()
    XMLData = DataSet1
End Function

```

```
End Class
```

Листинг 13.2.

Если содержимое XML-файла, используемого в нашем примере, будет выглядеть следующим образом:

```

<Document>
<Row><Document>
<Row>
  <Column1>1</Column1>
  <Column2>2</Column2>
  <Column3>Text1</Column3>
  <Column4>Text2</Column4>
</Row>
<Row>
  <Column1>3</Column1>
  <Column2>4</Column2>
  <Column3>Text3</Column3>
  <Column4>Text4</Column4>
</Row>
</Document>,
  <Column1>1</Column1>
  <Column2>2</Column2>
  <Column3>Text1</Column3>
  <Column4>Text2</Column4>
</Row>
<Row>

```

```

    <Column1>3</Column1>
    <Column2>4</Column2>
    <Column3>Text3</Column3>
    <Column4>Text4</Column4>
  </Row>
</Document>,
то в результате работы данного Web- сервиса мы получим:
<?xml version="1.0" encoding="utf-8" ?>
- <DataSet xmlns="http://tempuri.org/">
-       <xs:schema          id="Document"          xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema"  xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata">
-       <xs:element          name="Document"          msdata:IsDataSet="true"
msdata:Locale="ru-RU">
-       <xs:complexType>
-       <xs:choice maxOccurs="unbounded">
-       <xs:element name="Row">
-       <xs:complexType>
-       <xs:sequence>
        <xs:element name="Column1" type="xs:string" minOccurs="0" />
        <xs:element name="Column2" type="xs:string" minOccurs="0" />
        <xs:element name="Column3" type="xs:string" minOccurs="0" />
        <xs:element name="Column4" 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">
- <Document xmlns="">
-       <Row          diffgr:id="Row1"          msdata:rowOrder="0"
diffgr:hasChanges="inserted">
        <Column1>1</Column1>
        <Column2>2</Column2>
        <Column3>Text1</Column3>
        <Column4>Text2</Column4>
      </Row>
-       <Row          diffgr:id="Row2"          msdata:rowOrder="1"
diffgr:hasChanges="inserted">
        <Column1>3</Column1>
        <Column2>4</Column2>
        <Column3>Text3</Column3>

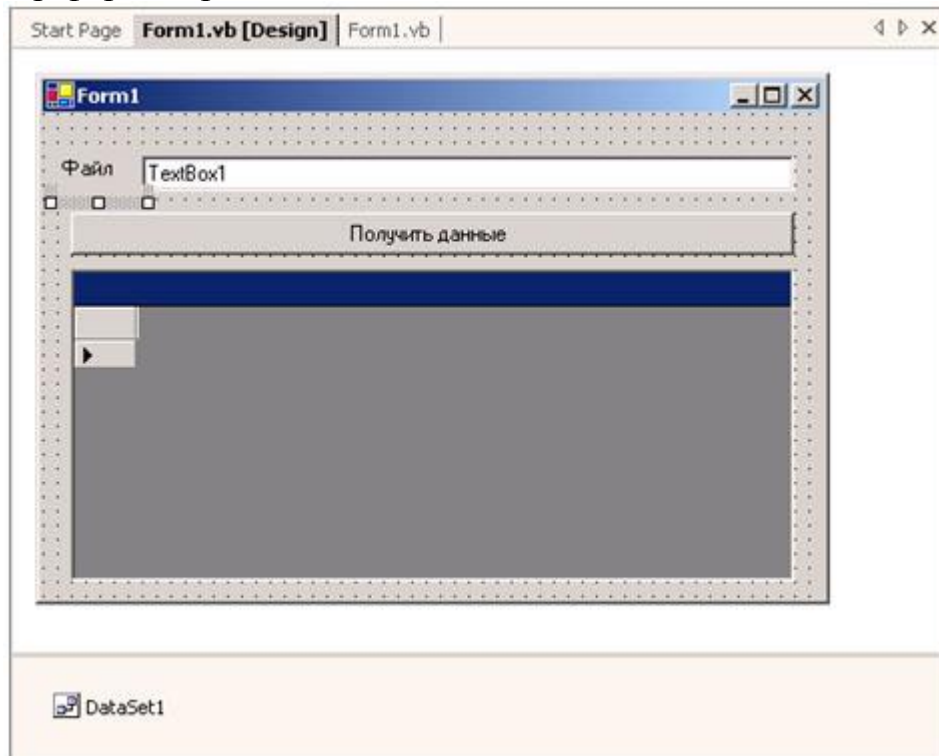
```

```
<Column4>Text4</Column4>
</Row>
</Document>
</diffgr:diffgram>
</DataSet>
```

Листинг 13.3.

Для создания клиентского приложения нужно для начала создать новый проект. Для этого как и в предыдущем примере следует выполнить команду меню File | New | Project. В появившемся диалоговом окне New Project в наборе Templates выбрать шаблон Windows Application и в поле Name указать наименование создаваемого приложения.

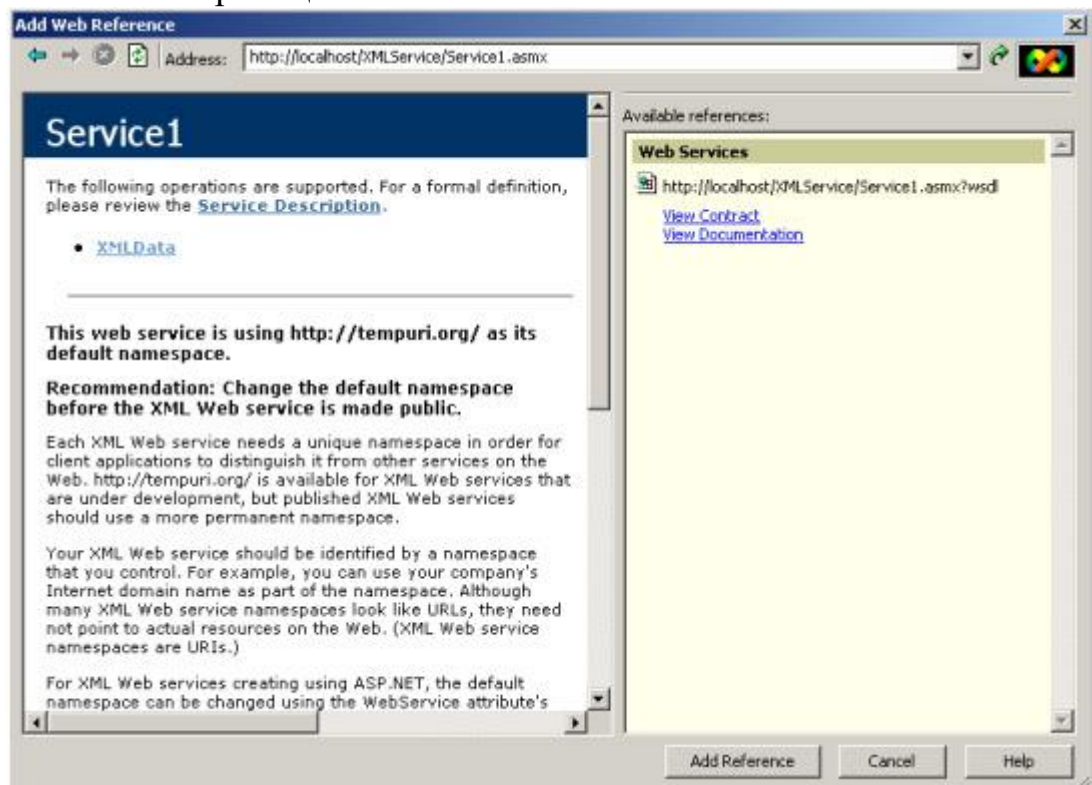
После этого Visual Studio .NET создаст форму с именем Form1, устанавливаемым по умолчанию. В окне этой формы нам потребуется разместить один элемент TextBox, одно текстовое поле Label, одну кнопку Button, один компонент DataGridView, и один компонент DataSet с вкладки Data. Пример формы приведен ниже.



После этого следует установить связь с искомым Web-сервисом. Для этого надо в окне Solution Explorer выбрать наименование создаваемого приложения и правым щелчком мыши вызвать для него контекстное меню. В этом контекстном меню необходимо выполнить команду Add Web Reference, после чего будет активизировано одноименное диалоговое окно, позволяющее установить ссылки на используемые сервисы.

Для того чтобы получить ссылки на функции Web-сервиса и перенести их в проект, нужно в текстовом поле Address указать URL необходимого сервиса и загрузить ресурс, располагающийся по этому адресу. После этого в

левой части искомого окна Add Web Reference будет отображено содержимое стартовой Web-страницы.



После того как искомые ссылки на функции Web-сервиса будут найдены, при помощи кнопки Add Reference их необходимо добавить к разрабатываемому проекту. После чего можно обращаться к функциям Web-сервиса в приложении (функция Button_Click).

```
Public Class Form1
```

```
    Inherits System.Windows.Forms.Form
```

```
#Region " Windows Form Designer generated code "
```

```
Public Sub New()
```

```
    MyBase.New()
```

```
    'This call is required by the Windows Form Designer.
```

```
    InitializeComponent()
```

```
    'Add any initialization after the InitializeComponent() call
```

```
End Sub
```

```
'Form overrides dispose to clean up the component list.
```

```
Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
```

```
    If disposing Then
```

```
        If Not (components Is Nothing) Then
```

```

        components.Dispose()
    End If
End If
MyBase.Dispose(disposing)
End Sub

```

'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Windows Form Designer

'It can be modified using the Windows Form Designer.

'Do not modify it using the code editor.

Friend WithEvents TextBox1 As System.Windows.Forms.TextBox

Friend WithEvents Label1 As System.Windows.Forms.Label

Friend WithEvents Button1 As System.Windows.Forms.Button

Friend WithEvents DataGridView1 As System.Windows.Forms.DataGridView

Friend WithEvents DataSet1 As System.Data.DataSet

<System.Diagnostics.DebuggerStepThrough()> Private Sub

InitializeComponent()

Me.TextBox1 = New System.Windows.Forms.TextBox()

Me.Label1 = New System.Windows.Forms.Label()

Me.Button1 = New System.Windows.Forms.Button()

Me.DataGridView1 = New System.Windows.Forms.DataGridView()

Me.DataSet1 = New System.Data.DataSet()

CType(Me.DataGridView1,

System.ComponentModel.ISupportInitialize).BeginInit()

CType(Me.DataSet1,

System.ComponentModel.ISupportInitialize).BeginInit()

Me.SuspendLayout()

,

'TextBox1

,

Me.TextBox1.Location = New System.Drawing.Point(56, 24)

Me.TextBox1.Name = "TextBox1"

Me.TextBox1.Size = New System.Drawing.Size(224, 20)

Me.TextBox1.TabIndex = 0

Me.TextBox1.Text = "TextBox1"

,

'Label1

,

Me.Label1.Location = New System.Drawing.Point(8, 24)

Me.Label1.Name = "Label1"

Me.Label1.Size = New System.Drawing.Size(48, 23)

Me.Label1.TabIndex = 1

```

Me.Label1.Text = "Файл"
,
'Button1
,
Me.Button1.Location = New System.Drawing.Point(16, 56)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(264, 24)
Me.Button1.TabIndex = 2
Me.Button1.Text = "Получить данные"
,
'DataGrid1
,
Me.DataGrid1.DataMember = ""
Me.DataGrid1.DataSource = Me.DataSet1
Me.DataGrid1.HeaderForeColor =
System.Drawing.SystemColors.ControlText
Me.DataGrid1.Location = New System.Drawing.Point(16, 88)
Me.DataGrid1.Name = "DataGrid1"
Me.DataGrid1.Size = New System.Drawing.Size(264, 80)
Me.DataGrid1.TabIndex = 3
,
'DataSet1
,
Me.DataSet1.DataSetName = "NewDataSet"
Me.DataSet1.Locale = New System.Globalization.CultureInfo("ru-RU")
,
'Form1
,
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.ClientSize = New System.Drawing.Size(292, 273)
Me.Controls.AddRange(New System.Windows.Forms.Control()
{Me.DataGrid1, Me.Button1, Me.Label1, Me.TextBox1})
Me.Name = "Form1"
Me.Text = "Form1"
CType(Me.DataGrid1,
System.ComponentModel.ISupportInitialize).EndInit()
CType(Me.DataSet1,
System.ComponentModel.ISupportInitialize).EndInit()
Me.ResumeLayout(False)

End Sub

#End Region

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim xml = New WindowsApplication2.localhost.Service1()
    DataSet1.Merge(xml.XMLData(TextBox1.Text))
End Sub
End Class

```

После запуска приложение будет выглядеть следующим образом:

The screenshot shows a Windows application window titled "Form1". At the top, there is a label "Файл" followed by a text box containing "1.xml". Below this is a button labeled "Получить данные". Under the button is a data grid with a blue header bar. The grid has four columns labeled "Column1", "Column2", "Column3", and "Column4". The first row of data contains the values "1", "2", "Text1", and "Text2". The second row contains "3", "4", "Text3", and "Text4". The third row is marked with an asterisk "*" in the first column, indicating it is a new or modified record. The grid also includes a small navigation icon in the top right corner.

	Column1	Column2	Column3	Column4
▶	1	2	Text1	Text2
	3	4	Text3	Text4
*				

Контрольные вопросы:

1. Web- службы. Основные понятия. Назначение.
2. История развития Web-служб. Модель COM/DCOM. .Стандарт CORBA/IIOP. Недостатки перечисленных технологий.
3. Web-службы .Net. Достоинства. Архитектура