

Tema – Servicii Web

1. Sa se creeze un serviciu Web care sa contina urmatoarele metode :
 - a) Conversie F to C si invers;
 - b) Afisarea orei si datei exacte;
 - c) Afisarea unei liste cu 5 elemente;
 - d) Conversia Intre doua valute (ex: lei - euro).
2. Sa se creeze un client de tip Windows Form Application care sa utilizeze metodele serviciului.

1. Interfata clientului de tip Windows Form:

Form1

View List

Data 11.04.2024

Ora 19:04

Temperatura C

Temperatura F

Rezultat conversie

F to C

C to F

Ron to Euro

Convert Ron To Euro

→ afisarea listei:

Form1

Tricou
Bluza
Pantaloni
Camasa
Papuci

View List

Data11.04.2024

Ora19:04

Temperatura C

Temperatura F

Rezultat conversie

F to C

C to F

Ron to Euro

Convert Ron To Euro

→ Conversia celor doua temperaturi:

Form1

Tricou
Bluza
Pantaloni
Camasa
Papuci

View List

Data11.04.2024

Ora19:04

Temperatura C25

Temperatura F

Rezultat conversie77

F to C

C to F

Ron to Euro

Convert Ron To Euro

Tricou
Bluza
Pantaloni
Camasa
Papuci

View List

Data

11.04.2024

Ora

19:04

Temperatura C

25

Temperatura F

77

Rezultat conversie

25

F to C

C to F

Ron to Euro

Convert Ron To Euro

→ Conversia monedei:

Tricou
Bluza
Pantaloni
Camasa
Papuci

View List

Data

11.04.2024

Ora

19:04

Temperatura C

25

Temperatura F

77

Rezultat conversie

25

F to C

C to F

Ron to Euro

100

20,12 euro

Convert Ron To Euro

2. Codul aferent pentru serviciul web creat:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;

namespace Lucrarea_4
{
    /// <summary>
    /// Summary description for WebService1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    2 references
    public class WebService1 : System.Web.Services.WebService
    {
        [WebMethod]
        1 reference
        public double FahrenheitToCelsius(double temperatureFahrenheit)
        {
            return (temperatureFahrenheit - 32) * 5 / 9;
        }
    }
}
```

```
[WebMethod]
1 reference
public List<string> GetList()
{
    return new List<string> { "Tricou", "Bluza", "Pantaloni", "Camasa", "Papuci" };
}

[WebMethod]
1 reference
public double CelsiusToFahrenheit(double temperatureCelsius)
{
    return (temperatureCelsius * 9 / 5) + 32;
}
```

```

[WebMethod]
1 reference
public DateTime GetCurrentDateTime()
{
    return DateTime.Now;
}

[WebMethod]
1 reference
public double ConvertCurrencyToEuro(double amountInRON)
{
    return amountInRON / 4.97;
}
}

```

3. Codul aferent pentru clientul de tip Windows Forms:

```

1 using Lucrarea_4;
2 using System;
3 using System.Collections.Generic;
4 using System.ComponentModel;
5 using System.Data;
6 using System.Drawing;
7 using System.Linq;
8 using System.Text;
9 using System.Threading.Tasks;
10 using System.Windows.Forms;
11
12 namespace client_forms
13 {
14     3 references
15     public partial class Form1 : Form
16     {
17         private Lucrarea_4.WebService1 webServiceClient;
18
19         1 reference
20         public Form1()
21         {
22             InitializeComponent();
23             webServiceClient = new Lucrarea_4.WebService1();
24         }
25     }
26 }

```

```

24 1 reference
private void AfisareDataSiOra()
25 {
26     DateTime currentDateAndTime = webServiceClient.GetCurrentDateTime();
27     dataTextBox.Text = currentDateAndTime.ToShortDateString();
28     oraTextBox.Text = currentDateAndTime.ToShortTimeString();
29 }
30
31 1 reference
private void Form1_Load(object sender, EventArgs e)
32 {
33     AfisareDataSiOra();
34 }
35
36 1 reference
private void lista_SelectedIndexChanged(object sender, EventArgs e)
37 {
38 }
39
40 1 reference
private void temperaturaCTextBox_TextChanged(object sender, EventArgs e)
41 {
42 }
43 }

```

```

44 1 reference
private void adaugareListaButton_Click(object sender, EventArgs e)
45 {
46     List<string> items = webServiceClient.GetList();
47     lista.Items.Clear();
48     lista.Items.AddRange(items.ToArray());
49 }
50
51
52
53 1 reference
private void TempC_Click(object sender, EventArgs e)
54 {
55 }
56 }
57
58 1 reference
private void TempF_Click(object sender, EventArgs e)
59 {
60 }
61 }
62

```

```

63 private void tempFTextBox_TextChanged(object sender, EventArgs e)
64 {
65     .....
66 }
67
68 1 reference
69 private void rezConv_Click(object sender, EventArgs e)
70 {
71     .....
72 }
73 1 reference
74 private void rezultatConversieTextBox_TextChanged(object sender, EventArgs e)
75 {
76     .....
77 }
78 1 reference
79 private void FToCbutton_Click(object sender, EventArgs e)
80 {
81     if (double.TryParse(tempFTextBox.Text, out double temperatureFahrenheit))
82     {
83         double temperatureCelsius = webServiceClient.FahrenheitToCelsius(temperatureFahrenheit);
84         rezultatConversieTextBox.Text = temperatureCelsius.ToString();
85     }
86     else
87     {
88         MessageBox.Show("Te rog introdu o valoare valida pentru temperatura Fahrenheit.", "Eroare", MessageBoxButtons.OK);
89     }
90 }
91 1 reference
92 private void CToFbutton_Click(object sender, EventArgs e)
93 {
94     if (double.TryParse(temperaturaCTextBox.Text, out double temperatureCelsius))
95     {
96         double temperatureFahrenheit = webServiceClient.CelsiusToFahrenheit(temperatureCelsius);
97         rezultatConversieTextBox.Text = temperatureFahrenheit.ToString();
98     }
99     else
100     {
101         MessageBox.Show("Te rog introdu o valoare valida pentru temperatura Celsius.", "Eroare", MessageBoxButtons.OK);
102     }
103 }
104
105 1 reference
106 private void EuroToRonLabel_Click(object sender, EventArgs e)
107 {
108     .....
109 }
110
111 1 reference
112 private void sumaRonTextBox_TextChanged(object sender, EventArgs e)
113 {
114     .....
115 }
116 1 reference
117 private void SumaEuroTextBox_TextChanged(object sender, EventArgs e)
118 {
119     .....
120 }

```

```

119 private void euroToRonButton_Click(object sender, EventArgs e)
120 {
121     if (double.TryParse(SumaEuroTextBox.Text, out double euroAmount))
122     {
123         try
124         {
125             double convertedAmountInRON = webServiceClient.ConvertCurrencyToEuro(euroAmount);
126             sumaRonTextBox.Text = convertedAmountInRON.ToString("0.00 euro");
127         }
128         catch (Exception ex)
129         {
130             MessageBox.Show("Eroare la conversia sumei!" + ex.Message, "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Error);
131         }
132     }
133     else
134     {
135         MessageBox.Show("Te rog introdu o valoare valida pentru suma in euro.", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Error);
136     }
137 }
138

```

```

139 private void oraTextBox_TextChanged(object sender, EventArgs e)
140 {
141 }
142
143
144 private void Ora_Click(object sender, EventArgs e)
145 {
146 }
147
148
149 private void Data_Click(object sender, EventArgs e)
150 {
151 }
152

```

```

153
154 private void dataTextBox_TextChanged(object sender, EventArgs e)
155 {
156 }
157

```




WebService1

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [CelsiusToFahrenheit](#)
- [ConvertCurrencyToEuro](#)
- [FahrenheitToCelsius](#)
- [GetCurrentDateTime](#)
- [GetList](#)

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services created using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "<http://microsoft.com/webservices/>":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

```
<WebService(Namespace="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
};
```

For more details on XML namespaces, see the W3C recommendation on [Namespaces in XML](#).

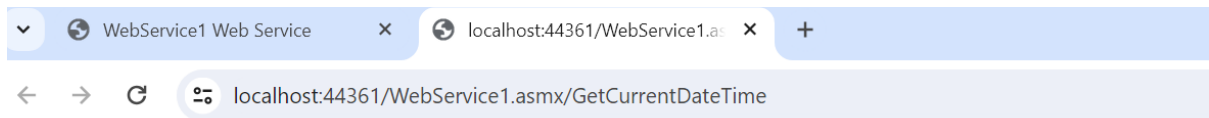
For more details on WSDL, see the [WSDL Specification](#).

For more details on URIs, see [RFC 2396](#).



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

```
<ArrayOfString xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://tempuri.org/">
  <script/>
  <string>Tricou</string>
  <string>Bluza</string>
  <string>Pantaloni</string>
  <string>Camasa</string>
  <string>Papuci</string>
</ArrayOfString>
```



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

```
<dateTime xmlns="http://tempuri.org/">
  <script/>
  2024-04-11T19:18:18.6773853+03:00
</dateTime>
```