Упражнение №4 по ПРС

REST

REST е съкратено Representational State Transfer (за това понякога се изписва като " ReST".) Тя разчита на клиент - сървър, кеширан, комуникационен протокол, като в почти всички случаи , се използва HTTP протокол.

REST е архитектурен стил за проектиране на мрежови приложения . Идеята е, че вместо да използват сложни механизми за свързване между машини (като CORBA , RPC или SOAP), прост HTTP се използва за заявки между машините. REST е архитектура, където всяка информация или понятие, която може да се обособи, се разглежда като ресурс и се идентифицира чрез идентификатор на ресурси URI.

- В много отношения , World Wide Web сама по себе си , основана на HTTP , може да се разглежда като REST архитектура. Общо казано, може да се гледа на интернет страница на като ресурс с ресурсен идентификатор URI = URL

RESTful приложения използват HTTP заявки за публикуване на данни (създаване и / или обновяване) , четене на данни (заявки), и изтриване на данни. По този начин REST използва HTTP за всички четири CRUD (Create/Read/Update/Delete) операции.

REST е лека алтернатива на механизми като RPC (Remote извикване на процедури) и уеб услуги (SOAP , WSDL , и др.) . Въпреки, че е прост , REST е пълнофункционален , всичко което може да се направи с Web Services , може да се направи и с RESTful архитектура.

REST не е " стандарт" . Никога няма да има W3C recommendataion за REST. Съществуват програмни рамки за REST , но REST е толкова прост, че най-често проектите създават "своя собствена" със стандартни функции (в езици като Perl , Java или C #).

Използвайки методологията на REST за уеб услуга включва използването на следните компоненти:

- **XML**: формат за представяне на ресурса.
- **HTTP**: методите GET, HEAD, POST, PUT и DELETE посочват какви действия да бъдат предприети.
- **URI**: URI, обикновено URL, е идентификатор на ресурс, който локализира уеб услугата (ресурса).

Създаване на REST услуга

1. Стартирайте Microsoft Visual Studio .NET.
2. **File --> New --> Project**. За езика C# изберете **WCF → WCF Service Application**.
3. В *.cs файлът към проекта (IService1.cs) добавете следния код (под коментара "// TODO: Add your service operations here"):

```
[OperationContract]
[WebInvoke(UriTemplate = "Addit?num1={num1}&num2={num2}",
```

```
        Method = "GET",
        BodyStyle = WebMessageBodyStyle.WrappedRequest,
        ResponseFormat = WebMessageFormat.Xml
        )
    ]
    string AddIt(int num1, int num2);
```

4. В *.cs файлът на услугата (Service1.svc.cs) добавете следния метод на класа `Service1`:

```
    public string AddIt(int num1, int num2)
    {
        return Convert.ToString(num1 + num2);
    }
```

5. В Web конфигурациите на вашия проект (файла Web.config) добавете следните настройки:

   В частта <system.serviceModel>:

```
  <services>
    <service name="WcfService1.Service1" behaviorConfiguration="web2">
      <endpoint address="" binding="webHttpBinding"
contract="WcfService1.IService1" behaviorConfiguration="web"></endpoint>
    </service>
  </services>
```

   В частта <behaviors>:

```
  <endpointBehaviors>
    <behavior name="web">
      <webHttp />
    </behavior>
  </endpointBehaviors>
```

   В частта <serviceBehaviors>:

```
    <behavior name="web2">
      <serviceMetadata httpGetEnabled="true" httpsGetEnabled="true" />
      <serviceDebug includeExceptionDetailInFaults="false" />
    </behavior>
```

6. Вече имате ваша уеб услуга, публикувана на локалния ви сървър. За да бъде разгледана и тествана я Натиснете F5. Разгледайте какво ви предлага публичния метод. Обърнете внимание на порта, на който е стартирана услугата.
   От браузър може да се извика публичния метод по следния начин:
   http://localhost:<port>/Service1.svc/AddIt?num1=1&num2=2
   http://localhost:<port>/Service1.svc/AddIt?num1=3&num2=5
   Какво трябва да стане: Връща се XML с резултата от функцията.

   Добавяне и обновяване на данни

7. В *.cs файлът към проекта (IService1.cs) добавете следния код:

```
        [OperationContract]
        [WebInvoke(UriTemplate = "getResource",
            Method = "GET",
            BodyStyle = WebMessageBodyStyle.WrappedRequest,
            ResponseFormat = WebMessageFormat.Xml
            )
        ]
        string getResource();

        [OperationContract]
        [WebInvoke(UriTemplate = "addResource?id={id}&value={value}",
            Method = "POST",
            BodyStyle = WebMessageBodyStyle.WrappedRequest,
            ResponseFormat = WebMessageFormat.Xml
            )
        ]
        string addResource(string id, string value);

        [OperationContract]
        [WebInvoke(UriTemplate =
"updateResource?id={id}&value={value}&isdel={isdel}",
            Method = "POST",
            BodyStyle = WebMessageBodyStyle.WrappedRequest,
            ResponseFormat = WebMessageFormat.Xml
            )
        ]
        string updateResource(string id, string value, bool isdel = false);
```

8. Създайте XML файл със следната структура:

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

</root>
```

9. В *.cs файлът на услугата (Service1.svc.cs) добавете следното поле на класа `Service1`, съдържащо пътя към XML файла.

```
        private const string resource_filename = @"<...>\resource.xml";
```

Например:
```
private const string resource_filename =
      @"C:\Users\student\Documents\resource.xml";
```

10. В *.cs файлът на услугата (Service1.svc.cs) добавете следните namespaces:

```
        using System.Xml;
```

11. В *.cs файлът на услугата (Service1.svc.cs) добавете следните методи на класа `Service1`:

```
        public string getResource()
        {

            XmlDocument doc = new XmlDocument();
```

```
            doc.Load(resource_filename);
            return doc.InnerXml;
        }

        public string addResource(string id, string value)
        {
            XmlDocument doc = new XmlDocument();
            doc.Load(resource_filename);

            XmlElement element;
            element = (XmlElement)doc.SelectSingleNode("//node()[@ID='" +
id + "']");
            if (element == null)
            {
                element =
(XmlElement)doc.DocumentElement.AppendChild(doc.CreateElement("p"));
                element.SetAttribute("ID", id);
                element.InnerText = value;
                doc.Save(resource_filename);
                return element.OuterXml;
            }
            else
            { return "Error"; }
        }

        public string updateResource(string id, string value, bool isdel =
false)
        {
            XmlDocument doc = new XmlDocument();
            doc.Load(resource_filename);

            XmlElement element;
            element = (XmlElement)doc.SelectSingleNode("//node()[@ID='" +
id + "']");

            if (element != null)
            {
                if (isdel)
                {
                    element.ParentNode.RemoveChild(element);
                }
                else
                {
                    if (element.FirstChild != null)
                    { element.RemoveChild(element.FirstChild); }

                    element.AppendChild(doc.CreateTextNode(value));
                }

                doc.Save(resource_filename);

                if (isdel)
                {
                    return "Deleted";
```

```
                }
                else
                {
                    return element.OuterXml;
                }
            }
            else
            { return "Error"; }
        }
```

12. Разгледайте новата услуга.
    От браузър може да се извика извличането на ресурса по следния начин:
    http://localhost:<port>/Service1.svc/getResource
    Извикването на другите два метода изисква POST. Може да се извикат от бутон в HTML страница, например:

```
<form action="<URI>" method="post">
  <input type="submit" value="Do it" />
</form>
```

    по следния начин:
    http://localhost: <port>/Service1.svc/addResource?id=1&value=test
    http://localhost: <port>/Service1.svc/updateResource?id=1&value=retest
    http://localhost: <port>/Service1.svc/updateResource?id=1&value=test&isdel=true
    REST приложение може да се тества и със специализирани инструменти, например RESTClientплъгина за Firefox (може да се намери на http://restclient.net/)
    Какво трябва да стане: Създадени са три публични метода. Първия връща целия ресурсен файл, другите добавят/променят/изтриват елемент от него по ID.

Използване на уеб услугата в ASP.NET страница

13. Стартирайте второ Microsoft Visual Studio .NET.
14. Създайте нов проект: **File→ New→ Project**. За езика C# изберете **Web → ASP.NET Web Forms Application**.
15. Заменете страницата Default.aspx със следния код:

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="WebApplication2._Default" %>

<asp:Content runat="server" ID="FeaturedContent"
ContentPlaceHolderID="FeaturedContent">
    <p>
        <em>ID</em>:
        <asp:TextBox ID="txtId" runat="server" Width="43px">1</asp:TextBox>
    </p>
    <p>
        <em>Value</em>:
        <asp:TextBox ID="txtValue" runat="server"
Width="440px">test</asp:TextBox>
    </p>
    <p>
        <em>Returned Result</em> :
        <asp:Literal ID="Result" runat="server">Literal</asp:Literal>
```

```
    </p>
    <p>
        <asp:Button ID="Button" OnClick="runGet_Click" runat="server"
Text="Get All">
        </asp:Button>
        <asp:Button ID="Button3" OnClick="runAdd_Click" runat="server"
Text="Add">
        </asp:Button>
        <asp:Button ID="Button1" OnClick="runUpdate_Click" runat="server"
Text="Update">
        </asp:Button>
        <asp:Button ID="Button2" OnClick="runDelete_Click" runat="server"
Text="Delete">
        </asp:Button>

</asp:Content>
```

16. В *.cs файлът на страницата (Default.aspx.cs) добавете следните namespaces:

```
using System.Xml;
using System.Net;
using System.IO;
```

17. В *.cs файлът на страницата (Default.aspx.cs) добавете следния код в класа _Default:

```
        protected const string serverpath =
"http://localhost:<port>/Service1.svc/";

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void runGet_Click(Object sender, EventArgs e)
        {
            string url = serverpath + "getResource";
            XmlDocument dom = new XmlDocument();
            dom.Load(url);

            Result.Text = dom.InnerXml;
        }

        protected void runAdd_Click(Object sender, EventArgs e)
        {
            string data = "addResource?id=" + txtId.Text + "&value=" +
txtValue.Text;
            string url = serverpath;

            WebRequest request = WebRequest.Create(url + data);
            request.Method="POST";
            request.ContentLength = 0;
            WebResponse response = request.GetResponse();
```

```
            StreamReader reader = new
StreamReader(response.GetResponseStream());
            Result.Text = reader.ReadToEnd();

            reader.Close();
            response.Close();

        }

        protected void runUpdate_Click(Object sender, EventArgs e)
        {
            string data = "updateResource?id=" + txtId.Text + "&value=" +
txtValue.Text;
            string url = serverpath;

            WebRequest request = WebRequest.Create(url + data);
            request.Method = "POST";
            request.ContentLength = 0;
            WebResponse response = request.GetResponse();

            StreamReader reader = new
StreamReader(response.GetResponseStream());
            Result.Text = reader.ReadToEnd();

            reader.Close();
            response.Close();
        }

        protected void runDelete_Click(Object sender, EventArgs e)
        {
            string data = "updateResource?id=" + txtId.Text + "&value=" +
txtValue.Text + "&isdel=true";
            string url = serverpath;

            WebRequest request = WebRequest.Create(url + data);
            request.Method = "POST";
            request.ContentLength = 0;
            WebResponse response = request.GetResponse();

            XmlDocument dom = new XmlDocument();
            dom.Load(response.GetResponseStream());
            Result.Text = dom.InnerXml;

            response.Close();
        }
```

18. Може да наблюдавате резултатите като стартирате приложението с F5.
    Какво трябва да стане: Страницата се стартира в браузър. Текста на страницата се променя в зависимост от резултатът върнат от услугите.

19. **Допълнителна задача** - ще ви я каже асистента.