

Daugiau apie Webservice

Anksčiau aprašėme, [kaip sukurti paprastą Webservice](#), o taip pat kaip jį [panaudoti ASP puslapiuose](#). Dabar šiek tiek išplėsime Webservice galimybių aprašymą. Tam tikslui šiek tiek pakeisime ir mūsų naudotą *Webservice*. Dabar jo kodas bus (kurį toliau ir nagrinėsime):

```
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.ComponentModel
Imports System.EnterpriseServices

<System.Web.Services.WebService(Namespace:="http://mysite/xml", _
    Description:="Labas, pasauli!", Name:="ManoLabas")> _
<System.Web.Services.WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
    <ToolboxItem(False)> _
Public Class Service1
    Inherits System.Web.Services.WebService

    <WebMethod(Description:="Pasisveikinimo teksto gražinimas, įterpiant agento ir pranešimo tekstus", _
        BufferResponse:=True, CacheDuration:=60, _
        EnableSession:=True, MessageName:="LabasPasauli")> _
    Public Function SayHelloWorld(ByVal Agent As String, ByVal Message As String) As String

        If Session("LabasSkaitliukas") Is Nothing Then
            Session("LabasSkaitliukas") = 1
        Else
            Session("LabasSkaitliukas") = CLng(Session("LabasSkaitliukas")) + 1
        End If

        Return "Hello " & Agent & vbCrLf & Message & vbCrLf & _
            "Kreipinys:" & Session("LabasSkaitliukas")
    End Function

    ' ----- Antrasis metodas
    <WebMethod(Description:="Tranzakcijos atributo panaudojimo pavyzdys", _
        TransactionOption:=TransactionOption.RequiresNew)> _
    Public Function DoSomeTransaction() As String
        ContextUtil.SetComplete()
        Return ContextUtil.TransactionId.ToString()
    End Function
End Class
```

1. Visų pirma, aprašydami *Webservice*, nurodome Jo aprašymą (*Description*) ir pavadinimą (*Name*)

Aprašymas (*Description*) – tai *Webservice* aprašantis tekstas, kuris galimiems jo naudotojams, kai generuojami aprašomieji dokumentai: aprašas arba pagalba.

Pavadinimas (*Name*) – *Webservice* pavadinimas. Nutylimoji reikšmė yra servisą realizuojančios klasės pavadinimas. Su WSDL sugeneravus serviso aprašą, pavadinimo atributas nurodo lokaliąją serviso pavadinimo dalį. Pavadinimas pateikiamas ir serviso pagalbos puslapyje.

Pastaba: *Webservice* pagalbos puslapis pateikiamas, kai iškviečiamas **.asmx** puslapis, nenurodžius nei metodo, nei jo parametrų, pvz.,

http://mySite/WS_Tst/Service1.aspx

2. Antra, *WebService* metodo aprašyme panaudojome atributus

Pirmasis atributas: **Description**

Šis atributas aprašo metodą ir jo tekstas yra pateikiamas *WebService* pagalbos puslapyje.

Antrasis atributas: **BufferResponse**.

Šis atributas leidžia valdyti *WebService* atsakymų buferizaciją ir gali įgauti *True* arba *False* reikšmes. Kai priskirta *True* reikšmė, *WebService* sukaupia visą atsakymą, prieš išsiunčiant jį. Tai leidžia padidinti efektyvumą, nes sumažina komunikacijų kiekį tarp veikiančio proceso ir IIS proceso. Nustačius *False*, atsakymas atidavinėjamas porcijomis (po 16 KB) – tai gali būti naudinga tik tada, kai atmintyje nenorime talpinti viso atsakymo, pvz., visos duomenų bazės kopijos. Todėl ir nutylimoji reikšmė yra *True*.

Trečiasis atributas: **CacheDuration**

Šis atributas leidžia nurodyti, kiek sekundžių *WebService* kešuos užklauso atsakymus. Nurodžius 0 (ši reikšmė yra nutylimoji), kešavimas yra išjungiamas. Kešavimas leidžia padidinti dažnai naudojamų užklauso efektyvumą, tačiau užima daugiau serverio resursų ir netinka tuo atveju, kai duomenys keičiasi labai dažnai.

Ketvirtasis atributas: **EnableSession**

Šis atributas leidžia valdyti sesijos būseną. Jo galimo reikšmės *True* ir *False*. Jei Sesijos būseną įjungta (*True*), *WebService* gali naudoti sesijos reikšmes per **Context.Current.Session** arba **WebService.Session**, jei ši savybė yra paveldėta iš *Webservice* pagrindinės klasės. Nutylimoji reikšmė yra *False*.

Mūsų išplėstame pavyzdyje, prieš gražinant rezultatą, sesijos kintamajame *LabasSkaitliukas* skaičiuojama, kiek **vienos sesijos metu** buvo kreipinių į *SayHelloWorld* metodą.

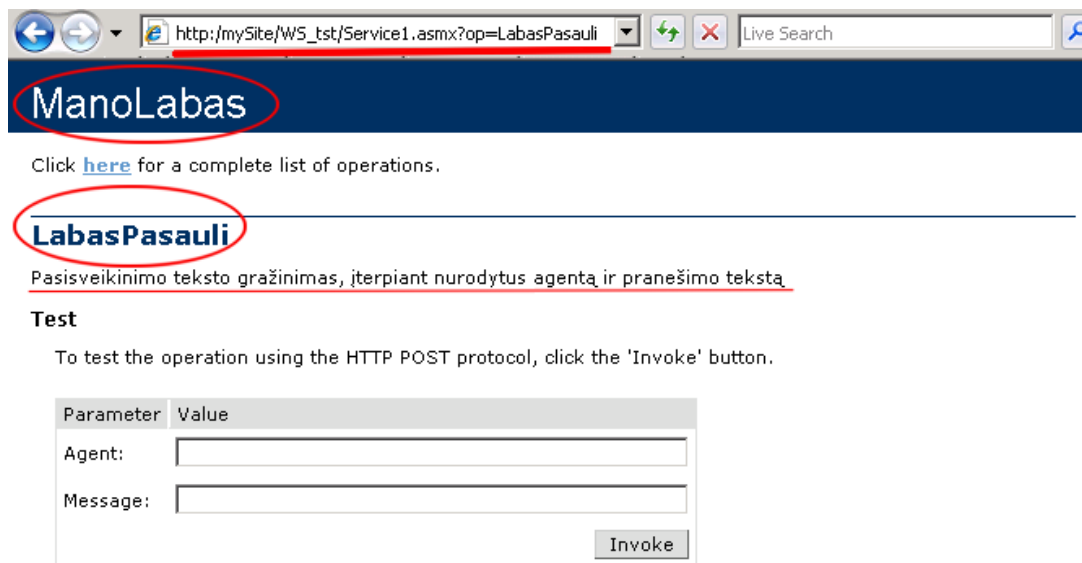
Penktasis atributas: **MessageName**

Šis atributas priskiria unikalų vardą (*alias*) metodui, turintį keletą kreipinio formatų (*overloaded*). Nutylimoji reikšmė – metodo pavadinimas. Kai *MessageName* yra panaudotas, SOAP atspindės šį vardą, o ne tikrąjį metodo pavadinimą. Mūsų pavyzdyje, metodą reikia kvieisti *LabasPasauli* vardu.

Šeštasis atributas: **TransactionOption**

Šis atributas leidžia *WebService* metodui veikti šakniniu transakcijos objektu. Nutylimoji reikšmė yra **TransactionOption.Disabled**. **Šikart** apie tai neišsiplėsimė; mūsų pavyzdyje realizuotas primityvus metodas, gražinantis transakcijos identifikatorių. Reikia atminti, kad šios galimybės panaudojimui reikia įtraukti nuorodą į **System.EnterpriseServices**

Iliustracija, pademonstruojanti *WebService* atributų reikšmes:



The screenshot shows a web browser window with the address bar containing the URL: `http://mySite/WS_tst/Service1.asmx?op=LabasPasauli`. The page has a dark blue header with the text "ManoLabas" circled in red. Below the header, there is a link "Click [here](#) for a complete list of operations." and another link "LabasPasauli" also circled in red. Underneath, there is a line of text: "Pasisveikinimo teksto gražinimas, įterpianč nurodytus agentą ir pranešimo tekstą". A section titled "Test" contains the instruction: "To test the operation using the HTTP POST protocol, click the 'Invoke' button." Below this is a form with two input fields: "Agent:" and "Message:". An "Invoke" button is located at the bottom right of the form.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /JS/Service1.asmx HTTP/1.1
```

WebService testavimas

[Ankstesniame straipsnyje](#) parodėme, kaip patikrinti *WebService* naudojantis IIS, o taip pat, kaip jį [kviešti iš ASP puslapio](#). Šįkart šį klausimą padetalizuosime.

Pirmiausia paminėsime, kaip galima gauti visą informaciją apie *WebService* naudojimą. Iškvieskite *WebService* taip:
`http://mySite/WS_Tst/Service1.asmx?WSDL`

Ši WSDL (*Web Service Definition Language*) užklausa pateiks visą informaciją, kurią galima naudoti dirbant su šiuo *WebService*. Mūsų parodomąjį *WebService* aprašas pateikiamas teksto gale (prie šios temos grįšime kitą kartą).

Tačiau nebūtina žinoti viską apie šį sugeneruotą „neskaitomą“ kodą. Testuoti *WebService* galime ir paprastuoju HTTP GET metodu. Tam naršyklėje nurodykite *WebService* metodą su jo parametrais. Mūsų pavyzdžiui, tai būtų, pvz.,
`http://mySite/WS_Tst/WS_Tst/Service1.asmx/LabasPasauli?Agent=Voras&Message=Nauduju`

Pastaba: naudojame *LabasPasauli*, o ne *SayHelloWorld* metodą, nes jį metodui priskyrėme naudodami *MessageName* atributą.

Tik, ... greičiausiai, gausite klaidą (vadinamąjį geltoną .Net puslapį), teigiančią, kad nežinomas užklauskos formatas (*Request format is unrecognized...*)

WebService dirba su GET, POST ir SOAP protokolais. Tačiau nuo .Net 1.1 nutylint yra „uždaryti“ GET ir POST protokolai. Taip padaryta saugumo sumetimais, Tačiau galime priversti naudoti tuos protokolus.

Tuo tikslu, atidarykime *WebService* **Web.config** konfigūracijų failą ir jame **system.web** grupėje suraskime (arba sukurkime, jei nėra **webServices** grupę, užpildydami, kaip parodyta žemiau:

```
<system.web>
  ...
  <webServices>
    <protocols>
      <add name="HttpGet"/>
    </protocols>
  </webServices>
</system.web>
```

Pastaba: Galite įtraukti ir kitus protokolus, jei jie reikalingi:

```
<add name="HttpPost"/>
<add name="HttpSoap"/>
<add name="HttpPostLocalhost"/>
```

Čia **HttpPostLocalhost** yra naujas protokolas, leidžiantis POST protokolą naudoti kviečiant *WebService* iš to paties serverio, kuriame yra *WebService*.

Išsaugojus šiuos pakeitimus, galėsime naudoti GET protokolą. Taigi, pažaiskime su nuoroda:

`http://mySite/WS_Tst/WS_Tst/Service1.asmx/LabasPasauli?Agent=Voras&Message=Nauduju`

Jį pateiks tokį *WebService* atsakymą:

```
Hello Voras Nauduju Kreipinys:1
```

Pirmiausia pastebėsime, kad keičiant **Message** parametro reikšmę, didėja po *Kreipinys* esantis skaičiukas (prisiminkime, kad tai kreipinių sesijos metu skaitliukas). Tačiau, pakartotinai įkėlus tą pačią nuorodą (*Refresh*, F5), skaitliukas nesikeičia. Kodėl?

Prisiminkime, kad pavyzdyje panaudojome keršavimo atributą (**CacheDuration:=60**), tad *WebService* visą minutę gražins užkešuoatą atsakymą. Todėl pagedaguokime mūsų *WebService* kodą, nurodydami **CacheDuration:=0**, ir iš naujo publikuokime naują *WebService* versiją. Pastebėsite, kad šįkart skaitliukas didėja ir atnaujinant puslapį!

Priedas: Mūsų WebService sugeneruotas WSDL tekstas

```
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://mySite/WS_Tst/xml" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://mySite/WS_Tst/xml" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Labas, pasauli!</wsdl:documentation>
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://mySite/WS_Tst/xml">
      <s:element name="LabasPasauli">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="Agent" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="Message" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="LabasPasauliResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="LabasPasauliResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="DoSomeTransaction">
        <s:complexType />
      </s:element>
      <s:element name="DoSomeTransactionResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="DoSomeTransactionResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="LabasPasauliSoapIn">
    <wsdl:part name="parameters" element="tns:LabasPasauli" />
  </wsdl:message>
  <wsdl:message name="LabasPasauliSoapOut">
    <wsdl:part name="parameters" element="tns:LabasPasauliResponse" />
  </wsdl:message>
  <wsdl:message name="DoSomeTransactionSoapIn">
    <wsdl:part name="parameters" element="tns:DoSomeTransaction" />
  </wsdl:message>
  <wsdl:message name="DoSomeTransactionSoapOut">
    <wsdl:part name="parameters" element="tns:DoSomeTransactionResponse" />
  </wsdl:message>
  <wsdl:portType name="ManoLabasSoap">
    <wsdl:operation name="SayHelloWorld">
      <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Pasisveikinimo teksto graūinimas, iterpiant nurodytus agenta
ir praneūimo teksta</wsdl:documentation>
      <wsdl:input name="LabasPasauli" message="tns:LabasPasauliSoapIn" />
      <wsdl:output name="LabasPasauli" message="tns:LabasPasauliSoapOut" />
    </wsdl:operation>
    <wsdl:operation name="DoSomeTransaction">
      <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Tranzakcijos atributo panaudojimo
pavyzdys</wsdl:documentation>
      <wsdl:input message="tns:DoSomeTransactionSoapIn" />
      <wsdl:output message="tns:DoSomeTransactionSoapOut" />
    </wsdl:operation>
  </wsdl:portType>

```

```
</wsdl:portType>
<wsdl:binding name="ManoLabasSoap" type="tns:ManoLabasSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="SayHelloWorld">
  <soap:operation soapAction="http://mySite/WS_Tst/xml/LabasPasauli" style="document" />
<wsdl:input name="LabasPasauli">
  <soap:body use="literal" />
</wsdl:input>
<wsdl:output name="LabasPasauli">
  <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="DoSomeTransaction">
  <soap:operation soapAction="http://mySite/WS_Tst/xml/DoSomeTransaction" style="document" />
<wsdl:input>
  <soap:body use="literal" />
</wsdl:input>
<wsdl:output>
  <soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ManoLabasSoap12" type="tns:ManoLabasSoap">
  <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="SayHelloWorld">
  <soap12:operation soapAction="http://mySite/WS_Tst/xml/LabasPasauli" style="document" />
<wsdl:input name="LabasPasauli">
  <soap12:body use="literal" />
</wsdl:input>
<wsdl:output name="LabasPasauli">
  <soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="DoSomeTransaction">
  <soap12:operation soapAction="http://mySite/WS_Tst/xml/DoSomeTransaction" style="document" />
<wsdl:input>
  <soap12:body use="literal" />
</wsdl:input>
<wsdl:output>
  <soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ManoLabas">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Labas, pasauli!</wsdl:documentation>
<wsdl:port name="ManoLabasSoap" binding="tns:ManoLabasSoap">
  <soap:address location="http://mySite/WS_Tst/Service1.asmx" />
</wsdl:port>
<wsdl:port name="ManoLabasSoap12" binding="tns:ManoLabasSoap12">
  <soap12:address location="http://mySite/WS_Tst/Service1.asmx" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```