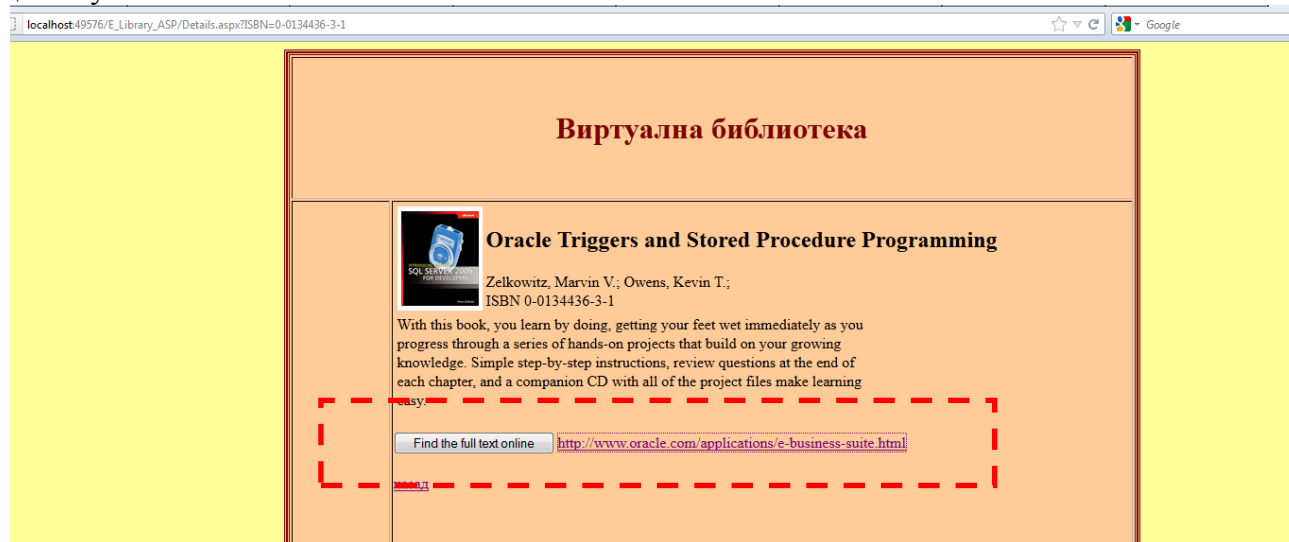


Упражнение 11

Фаза III-В. Преобразуване на eLibrary_ASP в потребител на уеб-услуга

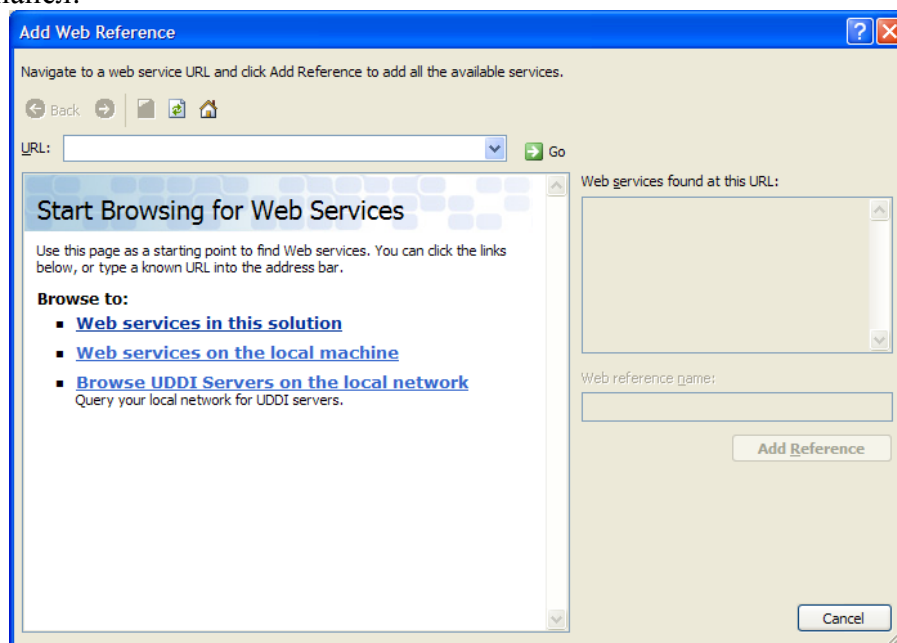
Цели

В сайта *eLibrary_ASP* към страницата за детайлна информация за книгата ще добавим един бутон, чрез който да се търси и показва на екрана URL за случая, когато е налична дадена книга в електронен вид. В базата данни *biblio.mdf* - използвана от сайта чрез *eLibraryServer* - не се съдържа такава информация. Ето защо, при натискане на новия бутон, сайтът ще се обърне към друг сайт – *eBookServices*, от който ще получи *URL* за дадената книга.



Стъпка 1

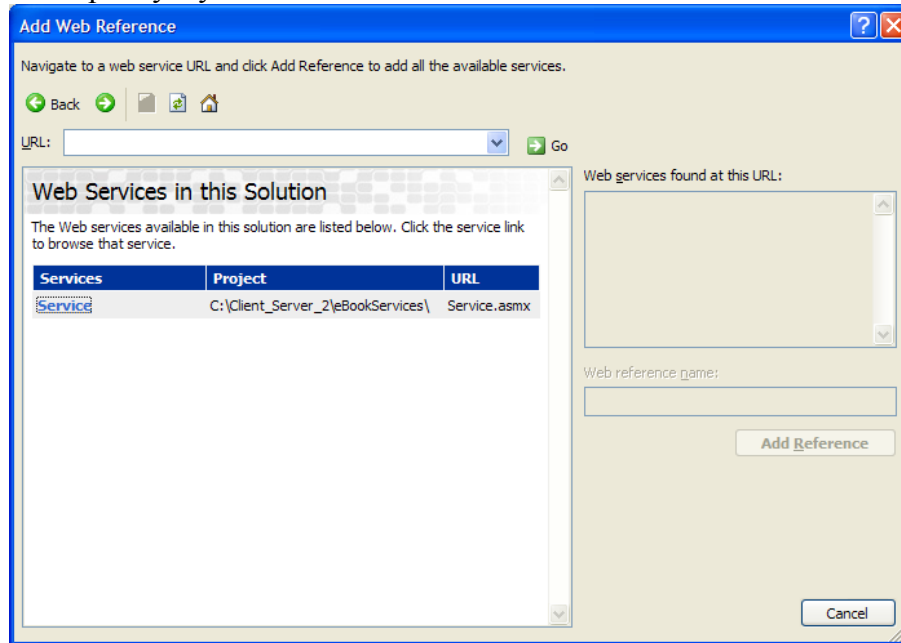
1 а) Отворете контекстното меню за сайта *eLibrary_ASP*, и изберете *Add Web Reference*. Отваря се следния панел:



1 б)

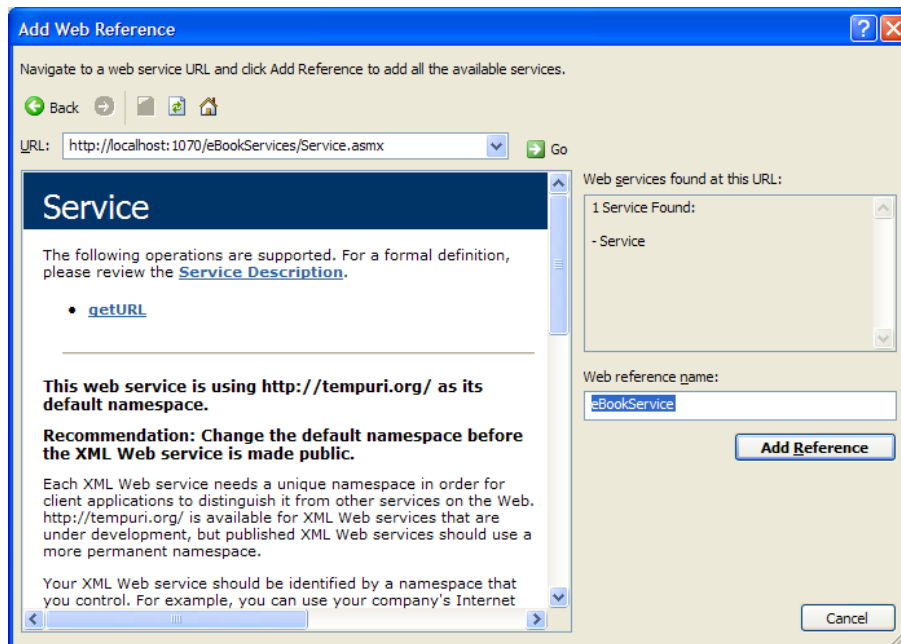
Изберете *Web services in this solution*.

Показва се панела за избор на услуги



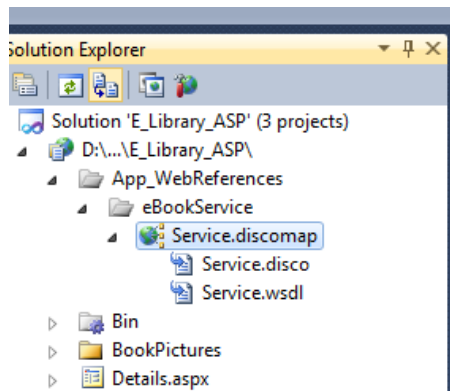
Изберете *Service*

В полето *web service name*: задайте името, с което ще наричаме този външен сайт доставящ услуги) когато се обръщаме към него в *eLibrary_ASP*. Трябва да е някакво по-говорящо име, а не предлаганото по подразбиране "*localhost*" – *eBookService*, например.



Натиснете бутона *Add Reference*

В резултат, в сайта е добавена папка *App_WebReferences* със следните файлове:



Стъпка 2

Преглед на файловете описващи сайта доставчик на уеб услуги

ЗАБЕЛЕЖКА: Тази стъпка не е задължителна. *Visual Studio* е прочело тези файлове вместо нас.

Използването на методите достъпни като уеб-услуги - в рамките на Visual Studio - не се различава от начина, по който използваме методите на който и да е друг външен клас.

Service.discomap

Service.discomap е помощен файл, който съдържа URL на който се намира два файла, описващи сайта доставчик на уеб услуги:

- Service.disco – DiscoveryDocumentReference – получава се от Service.asmx с параметър disco
в случая: `url="http://localhost:49421/Services/Service.asmx?disco"`
- Service.wsdl – ContractReference – получава се от Service.asmx с параметър wsdl
в случая: `url="http://localhost:49421/Services/Service.asmx?wsdl"`

Service.disco

Файлът *Service.disco* [`disco = discovery`] съдържа два URL адреса:

- адрес за извикване на услугите `< soap address= ...>`
- адрес на файла с WSDL описание на услугите `<contractRef ref= ...>`

В случая:

```
<contractRef ref="http://localhost:49421/Services/Service.asmx?wsdl"
docRef="http://localhost:49421/Services/Service.asmx"
xmlns="http://schemas.xmlsoap.org/disco/scl/" />
<soap address="http://localhost:49421/Services/Service.asmx"
xmlns:q1="http://tempuri.org/" binding="q1:ServiceSoap"
xmlns="http://schemas.xmlsoap.org/disco/soap/" />
```

Service.wsdl

Service.wsdl е файлът със стандартното WSDL описание на услугите от сайта eBookService. На фигурата са представени таговете на първо ниво.



Елементът **<definitions>** е главният (root) елемент в документа WSDL. Всички останали елементи се помещават вътре в него.

Елементът **<service>** задава името на web услугата. В този елемент се задават различните портове, на които е достъпна web услугата.

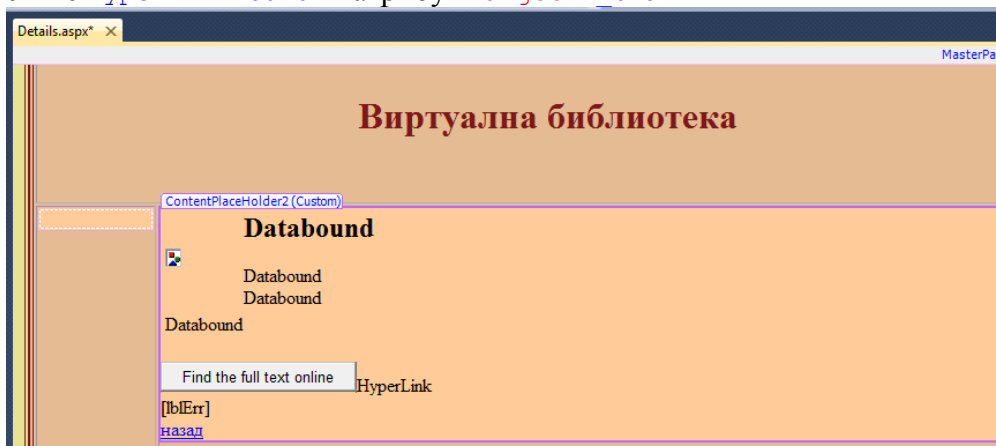
Елементът **<message>** отразява процеса на комуникация с услугата. Тъй като всяка услуга включва отправяне на заявка и получаване на отговор, то за всяка функция описана в WSDL (в случая само една getURL) трябва има по една двойка In и Out.

Елементът **<binding>** указва начинът, по който трябва са кодирани данните. В случая е указано, че се подчиняваме на спецификацията SOAP, която съдържа предварително дефиниран набор от правила за такива ситуации.

Стъпка 3

Отворете страницата **Details.aspx** в режим дизайн и добавете:

- бутон с име `btnFindURL`
- хиперлинк с име `HyperLinkFound` и атрибут `Target="_blank"`



```
<asp:Button ID="btnFindURL" runat="server" Text="Find the full text online"
    Width="170px" />
<asp:HyperLink ID="HyperLinkFound" runat="server"
    Target="_blank">HyperLink</asp:HyperLink>
```

За разлика от страницата в сайта eBookServices, тук няма входно поле за ISBN, защото се използва ISBN на книгата която е избрана в даден момент – и се съдържа в етикета с `id="ISBN"`.

```
<asp:Label id="ISBN" Runat="Server" Text='<%# "ISBN " & Eval("ISBN") %>' />
```

Стъпка 4

В *Details.aspx.cs* добавете функция **btnFindURL_Click** обработваща събитието натискане на бутона.

```
protected void btnFindURL_Click(object sender, EventArgs e)
{
    eBookService.Service oDAL=new eBookService.Service();
    String strURL ;
    try
    {
        strURL = oDAL.getURL(strISBN);
    }
    catch(Exception)
    {
        strURL = "No connection to Database";
    }
    if (strURL != "")
    {
        HyperLinkFound.Text = strURL;
        HyperLinkFound.NavigateUrl = strURL;
    }
    else
    {
        HyperLink1.Text = "Sorry, we have not found an e-book version!";
        HyperLink1.NavigateUrl = "";
    }
}
```

Стъпка 5

Сравнение на двете технологии – *Net Remoting* и *Web Services*

Нека да сравним двете функции в *Details.aspx.cs*:

- **btnFindURL_Click** и
- **Page_Load**

Във функцията **btnFindURL_Click** се дефинира обект *oDAL* от класа *eBookService.Service*

Във функцията **form1_Load** се дефинира обект *oDAL* от класа *eLibraryServer.DAL*

Спрямо уеб-формата, двата източника имат една и съща роля – доставят данни за визуализация; и два обекта се използват по еднообразен начин.

Има обаче **ограничения** върху **типовете данни**, които могат да се предават като параметри и резултативна стойност на функциите при уеб –услуги. Това могат да са само базовите типове данни. Обратно, при използване на Net remoting, където и клиентът и сървърът са приложения в среда на .NET , могат да се предават обекти от всеки клас, за които има дефинирана *сериализация*, т.е. могат да се предават *сложни обекти като Dataset*.

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Service : System.Web.Services.WebService
{
    [WebMethod]
    public String getURL(String pISBN)
    {
        String strReturn;
        strReturn = SearchInDB.getURL(pISBN);
        return strReturn;
    }
}
```

Публичният метод достъпен като уеб-услуга е *getURL()*. Той извиква вече създадения от нас метод *SearchInDB.getURL(pISBN)*.