

**T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



Veri Tabanı Laboratuvarı Dersi

Deney Föyü 5

Web Servisi Oluşturma

Aşağıda Windows ve Linux sistemler için web servisinin nasıl oluşturulduğu ve kullanıldığı ile ilgili bir derleme göreceksiniz. Öncelikle web servisi kavramını nasıl oluşturulduğunu bu makaleleri okuyarak anlamaya çalışın. (İnternette yer alan videolardan da faydalanabilirsiniz. Sadece bu videolara bağlı kalmadan kendi araştırmalarınızla da bilgi edinebilirsiniz)

C# ile web servis oluşturma : <http://www.youtube.com/watch?v=wsNY4VGtHPQ>

Java ile web servis oluşturma: <http://www.youtube.com/watch?v=o2Vjs8ylmFM>

Bu dökümanda size Visual Studio kullanarak web servisinin nasıl yapılacağı anlatılmıştır. Dilerseniz Java EE kullanarak ya da PHP kullanarak da aynı işlemleri yapabilirsiniz. Bu konuda özgürsünüz.

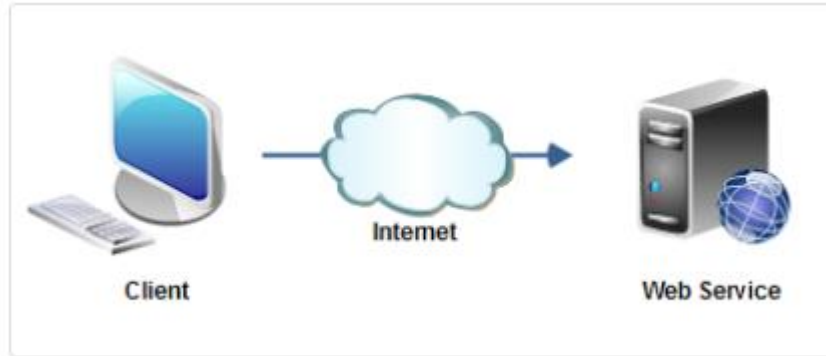
Konular:

- XML Web Servislerini Anlamak
- XML Web Servisleri Oluşturmak
- XML Web Servislerine İstemci Olmak

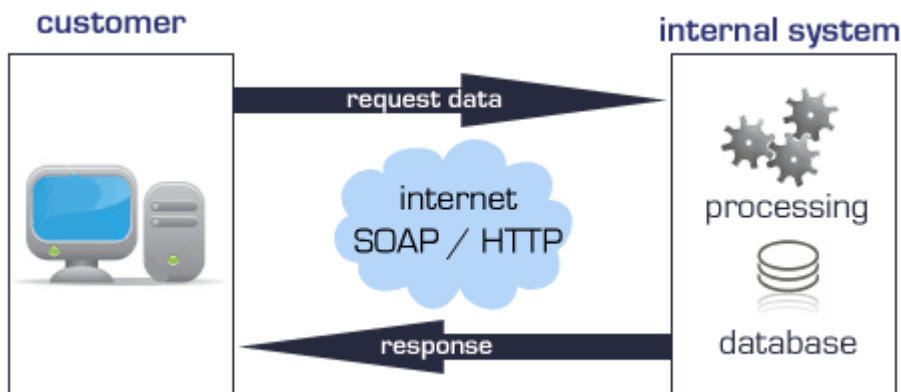
XML Web Servislerini Anlamak

Xml Web Servisleri yazılan program kodunu çalıştıran bileşenlerdir. Sıradan bileşenlerden farkı bu işlevi farklı bilgisayarlara da sunmasıdır. XML Web Servislerine farklı bir bilgisayar ve farklı bir platformdan istemci olunabilir.

Web servisler, HTTP protokolünü kullanarak hizmet sağlayan yapıların bütününe verilen isimdir.



Web servisler, istemci bilgisayar (client) tarafından gönderilen istekleri (request) yorumlayarak gönderilen isteğe göre istemciye yapısından bulundurduğu bilgileri aktarmaktadır.



Genel olarak web servisler SOAP (Simple Object Access Protocol/Basit Nesne Eriřim Protokolü) XML yapısını kullanarak çalışmaktadır. Bu yapının bizlere sağlamış olduđu avantaj platformdan bağımsız olarak istenilen yerden erişilmesini sağlamaktır.

Örneğin; .Net ile yazılan bir web servisi Javascript veya JSON ile ve birçok yazılım diliyle kullanabilmekteyiz. Bunun sebebi web servisten istediğimiz sonucun bizlere XML olarak sonuçları getirmesidir. Yani XML Web Servisleri farklı platformlar arasında ortak olarak kullanılabilen bileşenlerdir. XML Web Servisleri ile farklı platformlar arasında veri aktarımında belli standartlar sayesinde oldukça kolaylaşmaktadır.

XML Web Servisleri standart olan protokoller üzerinde çalışır. Bunlar, HTTP, XML, SOAP standartlardır. XML Web Servisleri adından da anlaşıldığı gibi mesaj alış verişı için XML standardını kullanır.

XML' in bir standart olması ve herhangi bir platforma bağı olmaması sayesinde XML Web Servisleri farklı platformların konuşmasını, veri alış verişini sağlar. Bu sayede uygulama geliştirme aşamasında programcılar farklı sistemleri kullanan uygulamalar yazarken standartları kullanarak işlerini daha kolay bir şekilde yapabilirler.

XML Web Servisleri HTTP protokolü üzerinden mesaj alış verişin yaptığı için internet üzerinden bir XML Web Servisine erişim yapılabilmektedir. Ek olarak bir intranetde de XML Web Servisi uygulaması yapılabilir. Önemli olan XML Web Servisinin HTTP protokolü ile yani bir web sunucusu ile sunuluyor olmasıdır. Tabi ki.NET Framework üzerinde programlanmış olan bir XML Web Servisinin sunulabilmesi için Internet Information Services (IIS) ve .NET Framework gerekmektedir.

XML Web Servislerinin Alt Yapısı

XML Web Servislerinin sağladığı faydalardan birisi XML Web Servisine istemci olacak ve o servisi kullanacak olan uygulamanın o servisin sadece adresini bilmesi yeterlidir. Bu şekilde servise eriştikten sonra XML Web Servisinin metodları çalıştırılabilir.

XML Web Servisleri Dizinleri

Yayında bulunan ve kaydedilmiş servisleri listeleyen merkezi noktalar, dizinler vardır. Bu dizinler sayesinde arama yapıp istediğiniz bir XML Web Servisine ulaşabilirsiniz. Bu dizinlerin XML Web Servislerini listelemesini belirleyen tanımlamalara Universal Description, Discovery and Integration (UDDI) denir.

Çeşitli XML Web Servisleri dizinleri mevcuttur. Bunlardan bir tanesi Microsoft'un XML Web Servisi dizini olan, <http://uddi.microsoft.com> sitesidir. Bu dizinlerde istediğiniz şekilde XML Web Servisi araması yapabileceğiniz gibi XML Web Servislerinizi de bu dizinlere kaydedebilirsiniz.

XML Web Servisleri Bulunması

Bu işlem ile istemciler XML Web Servislerini tanımlayan (WSDL) dokümanların konumlarını alırlar. Bu adım ile istemciler XML Web Servislerinin konumları ile ilgili bilgi sahibi olurlar.

XML Web Servisleri Tanımlaması

Xml Web Servisi Tanımlaması XML Web Servisinin biçimi hakkında bilgileri içeren bir XML dokümanıdır. Bu doküman ile XML Web Servisi üzerinde yapılabilecek olan işlemler istemciye bildirilir.

XML Web Servisinin İşleyişi

1. Eğer kullanılacak XML Web Servisinin adresi bilinmiyorsa bir UDDI dizini yardımı ile bir XML Web Servisi bulunur.
2. Discovery dosyasına (.disco) yapılan istek ile web servisinin tanımlama dosyasına yönlendirilir.
3. Web Servisinin açıklama dosyası olan WSDL dosyasına istek yapılır. Servis biçimi öğrenilir.
4. Artık XML Web Servisinin metodları çalıştırılmaya hazır olarak istemcinin hizmetindedir. XML Web Servisinin metodları çalıştırılır.

Bir istemci UDDI aracılığı ile bir XML Web Servisine eriştiğinde UDDI servisi istemciye XML Web Servisinin discovery dosyası olan .disco uzantılı dosyanın adresini döner. Bu .disco dosyası XML Web Servisini tanımlayan kaynaklara link içerir.

```
<?xml version="1.0" ?>
<disco:discovery
xmlns:disco="http://schemas.xmlsoap.org/disco"
xmlns:wSDL="http://schemas.xmlsoap.org/disco/wSDL">
<wSDL:contractRef
ref="http://www.siteniz.com/Hesapci.asmx?WSDL"/>
</disco:discovery>
```

İstemci .disco dokümanındaki bilgileri kullanarak sunucudan XML Web Servisi açıklama dosyasına istek yapar. Bu dosya WSDL dosyasıdır. WSDL (Web Services Description Language) XML tabanlı bir dosyadır ve istemcinin XML Web Servisinin işlevlerini öğrenip etkileşime geçebilmesini sağlar.

Bir istemci XML Web Servisinin metodlarını çalıştırması için gerekli olan işlemler şunlardır.

1. İstemci yazılım istemci tarafında kalmak üzere bir sınıf yapısı oluşturur. Bu sınıf XML Web Servisinin WSDL dosyasında tanımlanan yapısına uygundur. Yani XML Web Servisinde hizmete açılan sınıfın yapısı ile aynı yapıdadır. Buna aracı (proxy) nesne denir.
2. İstemci sunucu sınıfın bir metodunu çağırır.
3. İstemciye XML Web Servisi altyapısı metod çağırısını ve metodun parametrelerini SOAP mesajı olarak serialize eder. Ve bu mesajı XML Web Servisine yönlendirir.
4. Sunucudaki XML Web Servisi altyapısı gelen SOAP mesajını deserialize eder, nesnenin sunucu üzerinde bir örneğini oluşturur ve çağrı yapılan metoda parametreleri ile beraber çağrı yapar.
5. XML Web Servisi metodu çalıştırır ve geriye dönen değerleri altyapıya döndürür.

6. XML Web Servisi altyapısı dönecek değeri SOAP mesajı olarak serialize eder ve bu mesajı ağ üzerinden istemciye gönderir.
7. İstemci bilgisayardaki altyapı gelen SOAP mesajını deserialize eder ve değerleri aracı (proxy) nesneye aktarır.
8. Aracı (proxy) nesne ise aldığı değerleri kendi istemcisine aktarır.

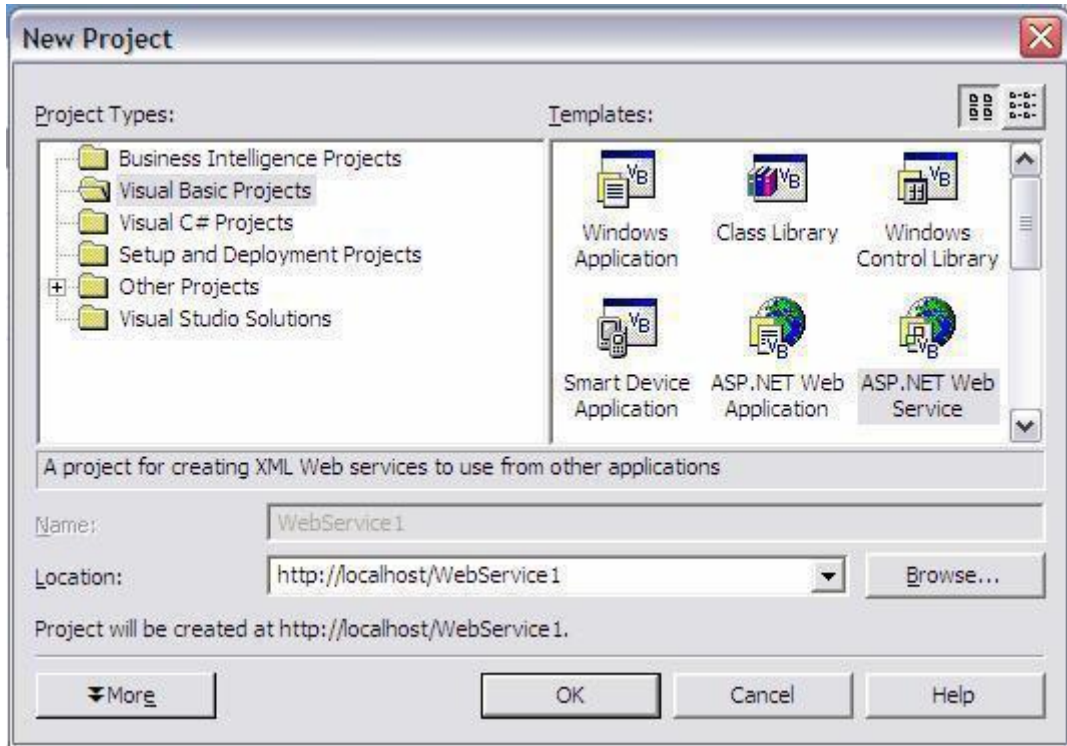
Bir XML Web Servisi oluşturabilmek için .NET Framework'ün bir bileşeni olan ASP.NET altyapısı kullanılır.

XML Web Servisleri Oluşturmak

XML Web Servisi oluşturmak bir bileşen oluşturmaktan farklı değildir. XML Web Servisleri ağ ortamının avantajı ağ ortamı üzerinden paylaşılabilir olmasıdır.

Visual Studio.NET ile Yeni Bir Web Servisi oluşturmak için:

1. File>New Project menü seçeneğine tıklayın.
2. Açılan New Project penceresinden uygulama geliştirmek istediğiniz dili seçtikten sonra ASP.NET Web Service seçeneğini seçerek OK butonuna basın.
3. Açılan projede .asmx uzantılı dosyalar web servisi uygulama dosyalarıdır.



Web Servis Projesinde Yer alan dosyalar

AssemblyInfo.vb : Bir assembly'nin versiyon bilgisi gibi bilgilerini düzenlemek için kullanılan dosyadır. Tüm .NET projelerinde yer alan bir dosyadır.

Global.asax ve Global.asax.vb : Global.asax ve onun codebehind dosyası olan Global.asax.vb dosyası ile uygulama ve oturum bazında olay yönetimi yapılabilir. Bu dosya uygulamanın ana klasöründe yer almalıdır.

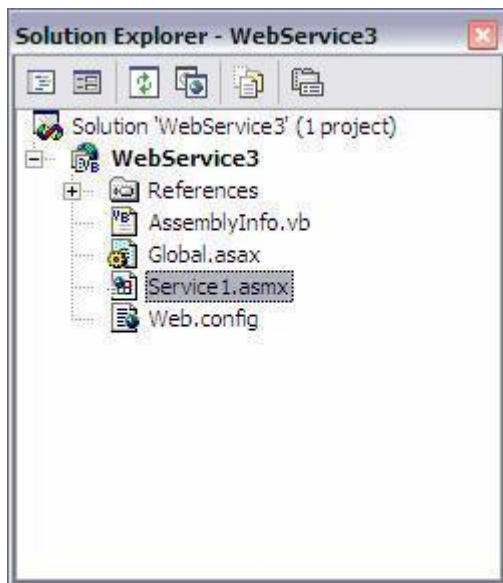
Service1.asmx ve Service1.asmx.vb : Bu iki dosya Service1.asmx ve codebehind dosyası olan

Service1.asmx.vb dosyaları bir Web Servisini oluştururlar. Service1.asmx'in görevi IIS tarafında Web Servisinin bulunabilmesini sağlamak ve Web Servisi için giriş noktası olmaktır.

Service1.asmx.vb dosyası yazılan program kodlarını içeren dosyadır. ASP.NET uygulamalarında olduğu gibi uygulama derlenince sonra oluşan .dll dosyasında bu kodlar yer alır.

Service1.asmx.vb dosyasının içeriği:

```
Public Class Service1
Inherits System.Web.Services.WebService
methodlar
methodlar
End Class
```



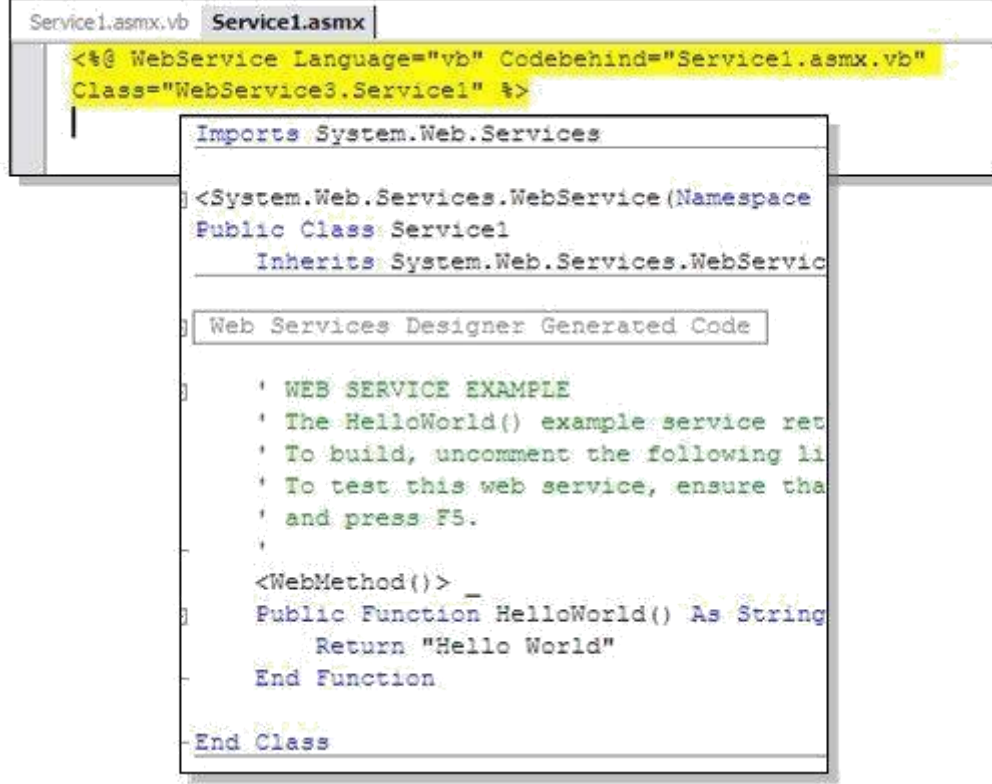
Web Servislerinin codebehind dosyaları System.Web.Services.WebService sınıfından miras alırlar.

Web Servisi içerisindeki metodun internet üzerinden sunulabilmesi <WebMethod()> özelliği ile nitelendirilmesi gerekmektedir.

```
<WebMethod()> Public Function HelloWorld() As String
    Return "Hello World"
End Function
```

Yukarıdaki kod bloğunda yazılan HelloWorld isimdeki fonksiyon web servisi üzerinden hizmete açılmış bir function tanımlamasıdır. Bu fonksiyonun web servisinden sunulmasını sağlayan ifade **<WebMethod(>** ifadesidir.

Yeni Proje Açtığınız da oluşturulan Web Servisi dosyasının içeriği şu şekildedir.



```
<?@ WebService Language="vb" Codebehind="Service1.asmx.vb"
Class="WebService3.Service1" %>

Imports System.Web.Services

<System.Web.Services.WebService (Namespace
Public Class Service1
    Inherits System.Web.Services.WebService

    Web Services Designer Generated Code

    ' WEB SERVICE EXAMPLE
    ' The HelloWorld() example service returns
    ' To build, uncomment the following line
    ' To test this web service, ensure that
    ' and press F5.

    <WebMethod(> _
    Public Function HelloWorld() As String
        Return "Hello World"
    End Function

End Class
```

Bu şekilde bir WebServisi tanımlamasını yapıp web üzerinden XML Web Servisi olarak sunulan bileşeninizi yaptığınızda aynı bilgisayar üzerinden servise <http://localhost/webservice1/service1.asmx> şeklinde erişim yaptığınız zaman XML Web Servis alt yapısı bir yardım sayfası sunacak ve servis üzerindeki metodları bu arabirim sayesinde listeleyp, test etmenizi sağlayacaktır. Bu sadece aynı bilgisayar üzerinden erişimlerde çalışır.

Ayrıca açılan bu yardım sayfasında Web Servisindeki metodların bilgilerini sunan SOAP biçiminde XML tanımlamaları da görülmektedir.

Servis Açıklama Sayfası

Bir XML Web Servisi için yapılan servis açıklamasına Web Services Description Language (WSDL) denir. Servis yardım sayfasında “Service Description” isminde bir link görülmektedir. Bu link servisinde WSDL açıklamalarına link vermektedir. Bu linkin yolu aşağıdaki şekildedir.

<http://sunucuadresi/servisdizini/service.asmx?WSDL> gerçek bir örneğine bakarsak localhost üzerindeki WebService2 içindeki Service1.asmx servisi için link şu şekilde olmalıdır:

<http://localhost/WebService2/Service1.asmx?WSDL>

Bu adrese erişebilmek için yardım sayfasındaki linki kullanmak zorunlu değildir. Bu adresi web tarayıcısının adres kısmına yazıp, sayfaya istek yaptığınız zaman aynı WSDL tanımlamasını görebilirsiniz. Tabiki adresinde sonuna ?WSDL querystring ifadesini eklemeniz gerekmektedir.

Yeni oluşturulmuş ve içinde sadece HelloWorld isminde metodumuz olan bir XML Web Servisi için WSDL XML dokümanı şu şekilde olacaktır.

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:s0="http://tempuri.org/WebService2/Service1"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
targetNamespace="http://tempuri.org/WebService2/Service1"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <s:schema elementFormDefault="qualified"
targetNamespace="http://tempuri.org/WebService2/Servi
cel">
      <s:element name="HelloWorld">
        <s:complexType />
      </s:element>
      <s:element name="HelloWorldResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="HelloWorldResult"
type="s:string" /> </s:sequence>
          </s:complexType>
        </s:element>
      </s:schema>
    </types>
    <message name="HelloWorldSoapIn">
      <part name="parameters"
element="s0:HelloWorld" /> </message>
    <message name="HelloWorldSoapOut">
      <part name="parameters"
element="s0:HelloWorldResponse" /> </message>
    <portType name="Service1Soap">
      <operation name="HelloWorld">
        <input message="s0:HelloWorldSoapIn" />
        <output message="s0:HelloWorldSoapOut" />
      </operation>
    </portType>
    <binding name="Service1Soap" type="s0:Service1Soap">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document" /> <operation name="HelloWorld">
```



```

    <soap:operation
soapAction="http://tempuri.org/WebService2/Service1/HelloWorld"
style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
<service name="Service1">
  <port name="Service1Soap" binding="s0:Service1Soap">
    <soap:address
      location="http://localhost/WebService2/Service1.asmx" />
  </port>
</service>
</definitions>
>

```

Bu dosyanın SOAP biçiminde düzenlenmiş bir XML dosyasıdır. XML Web Servisine daha sonra bir istemci olarak bağlanmak istediğinizde XML Web Servisi istemci altyapısı bu dosyadaki tanımlamalara bakarak bir aracı (proxy) sınıf oluşturacak ve istemci tarafından bu XML Web Servisine erişilebilmesini sağlayacaktır.

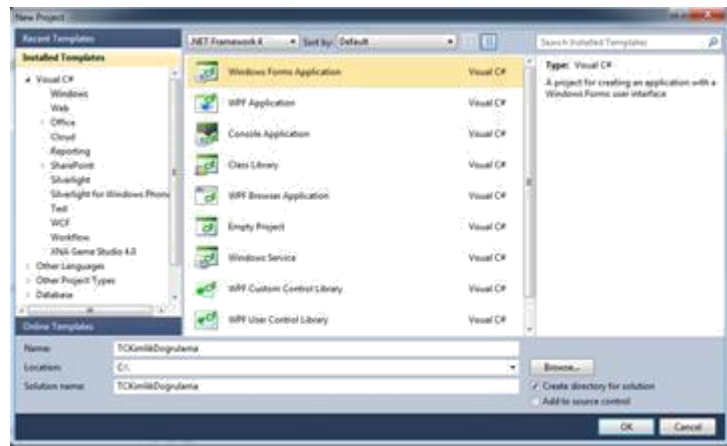
Bu aracı sınıfın oluşturulması işlemi Visual Studio.NET arabirimi ile referans verildiğinde yapılacaktır.

Fakat Visual Studio.NET kullanmadan bir XML Web Servisine istemci olmak istiyorsanız.NET Framework'ün bir parçası olan WSDL.exe komut satırı uygulamasını kullanabilirsiniz.

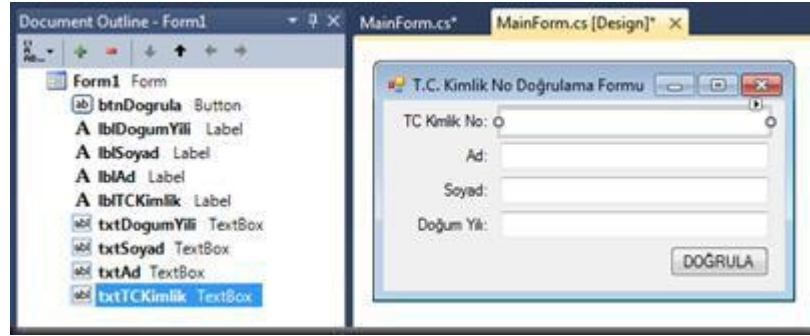
ÖRNEK: T.C. Kimlik Numarası Doğrulama - XML Web Servisi

Bu yazımızda, C# ile XML Web Servislerini kullanarak <http://tckimlik.nvi.gov.tr> adresinde hizmete sunulan vatandaşlık numarası doğrulama hizmetini ele alan bir örnek proje olsun.

“TCKimlikDogrulama” adında bir Windows Forms projesi oluşturarak başlayalım.



Proje açıldıktan sonra, doğrulama için web servisine gönderilecek olan bilgileri kullanıcıdan almak üzere aşağıdaki gibi bir form tasarımı yapılabilir.



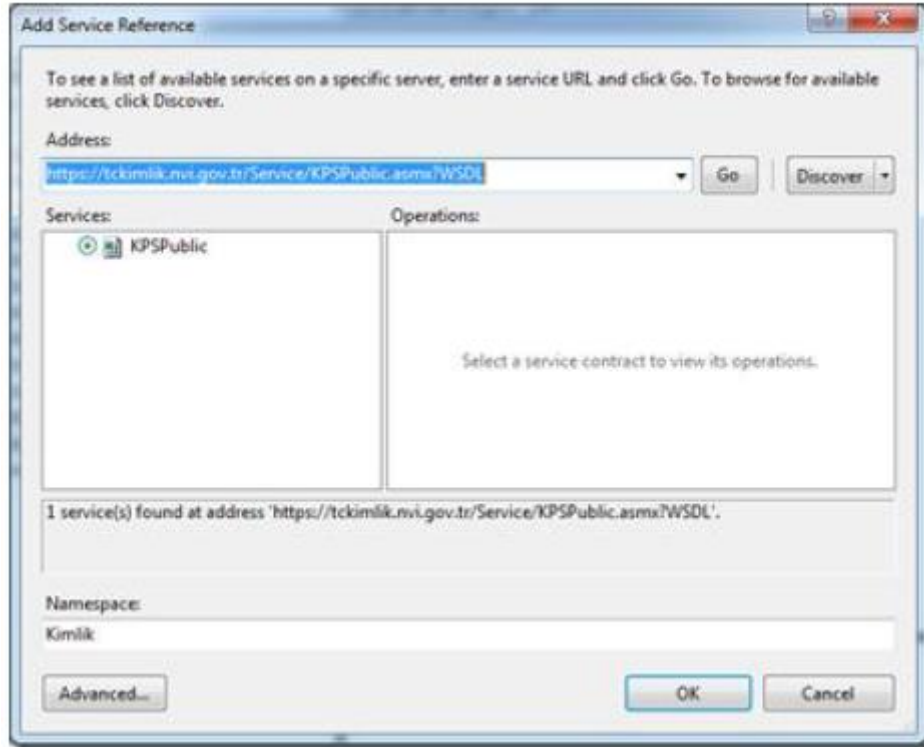
Servis referansını projeye eklemeye:

T.C. İçişleri Bakanlığı Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü tarafından kullanıma sunulan web servisinin adresini aşağıda bulabilirsiniz.

<https://tckimlik.nvi.gov.tr/Service/KPSPublic.asmx>

DİKKAT: KPS Servisi yalnızca SSL (https adresi) üzerinden hizmet veriyor.

Bu servisi proje içerisinde kullanabilmek için öncelikle **Solution Explorer** içerisinde yer alan **Services** i seçelim.



servisinin adresini
Namespace bilgisi
Services bölümünde

Bu işlemlerin ardından **OK** butonuna tıklayarak servis referansının projeye eklenme sürecini tamamlayalım. Şimdi de, servise çağrı yapıp sonucu alacak olan kodları projeye ekleyelim. Bunun için **btnDogrula** isimli buton kontrolünün Click olayına aşağıdaki metodu bağlayalım.

```

private void btnDogrula_Click(object sender, EventArgs e)
{
    long tckimlik = long.Parse(txtTCKimlik.Text);
    int dogumYili = int.Parse(txtDogumYili.Text);

    bool? durum;

    try
    {
        using (Kimlik.KPSPublicSoapClient servis = new Kimlik.KPSPublicSoapClient())
        {
            durum = servis.TCKimlikNoDogrula(tckimlik, txtAd.Text, txtSoyad.Text, dogumYili);
        }
    }
    catch
    {
        durum = null;
    }

    MessageBox.Show(durum.ToString());
}

```

Bu son işlemin ardından artık projeyi derleyip kullanmaya başlanılabilir.

ÖNEMLİ NOT: Servise doğrulama için gönderilecek ad ve soyad bilgileri tamamen büyük harflerle yazılmış olarak gönderilmelidir.

PHP ile Web Servisi Oluşturmak

Web servis ile internet üzerinde herhangi bir platformdaki, herhangi bir dille yazılmış bir uygulama ile veri alışverişi gerçekleştirebiliriz. Bunu sadece veri alışverişi olarak sınırlamak da pek doğru değil, ama girişte bunun böyle bilinmesi yeterli olacak.

Bir siteniz var ve bu sitenizden kullanıcılarınız üyelik sistemi ile yararlanabiliyor. Bu sistemde kullanıcılarınıza SMS ile bilgi vermeye yarayan bir özellik aktif etmek istiyorsunuz. Ama bu servisi kontrol etmek elimizde değil, bu yüzden farklı bir yerden satın almak zorundayız. Biz SMS servisini satın aldıktan sonra, satıcı bize SMS gönderme işlemi kendi uygulamamız içerisinde gerçekleştirebilmemiz için bir API (Application Programming Interface Uygulama Geliştirme Arayüzü) sunar. İşte bu API'lerin iletişim yöntemlerinden biri de web servislerdir.

Yukarıda da bahsedildiği gibi, farklı platformlar var ve bunların birbiri ile anlaşması gerekiyor. İşte burada da SOAP devreye giriyor. **SOAP** (Simple Object Access Protocol ,Basit Nesne Erişim Protokolü). Tabiki ne bu bahsettiğimiz API çağrıları için, ne de web servis işlemleri için tek protokol SOAP değildir. Örnekteki amaç SOAP dır.

PHP ile web servis geliştirme işlemi kolaydır. Bunu iki farklı yöntem ile gerçekleştirebiliriz. Birincisi WSDL yardımı ile ikincisi WSDL olmadan direkt PHP sınıfları ile. Birinci seçenek farklı bir yazımın konusu olacağından, ikinci yöntem; sistemin işleyebilmesi için bir sunucu bir de istemci olması gerekiyor. Her ikisinin farklı yerde farklı platform/dil de olması gerekiyordu, örneğin işlevselliğini gösterebilmek için.

İlk önce sunucu tarafındaki kodlar:

```

1 <?php
2 class SMS {

```

```

3     public function gonder($mesaj,$konu,$kime){
4         return 'mesaj gönderildi';
5     }
6 }
7 $sunucu=new SoapServer(null,array('uri'=>'http://ersindogan-
8 testuri'));
9
10 $sunucu->setClass('SMS');
11
12 $sunucu->handle();

```

Yukarıdaki kodun içeriği bir anlamda tam değil, sadece uzak yordam çağrısı yapılır ve bir sonuç döndürmesi yeterlidir. Bu yüzden SMS mesajını başarılı bir biçimde göndermiş gibi yapan bir sunucu oluşturulmuş olur. Evet birkaç satır kod ile bunu başardık.

SMS diye bir sınıf oluşturuldu, bu sınıf SMS göndermeye SMS gönderme işleminin sonucunu döndürmeye yarıyor. Sunucu olarak oluşturulan nesne bir **SoapServer** nesnesi. Bu SoapServer nesnesi, gelen uzak yordam çağrılarının, belirlenen şartlar doğrultusunda çalıştırılmasını ve

sonucunun istemciye döndürülmesini sağlamak ile görevli. Burada SoapServer nesnesi, SMS sınıfını baz almasını ve bu nesnenin metodlarını çalıştırabilir. Eğer bunun dışında çalıştırılması istenen sınıf-fonksiyon çağrısı gelirse SoapServer istemciye hata döndürecektir.

SoapSever nesnemizi burada WSDLsiz oluşturduk, WSDLsiz oluşturulan bir URI (Uniform Resource Identifier Tekörnek Kaynak Tanımlayıcı) tanımlamamız gerekir. Bu url herhangi bir benzersiz url olabilir. Daha sonra sunucunun baz alacağı nesne tanımlanır, son olarak da handle metodu ile de sunucumuzun gelen isteklere cevap vermesini sağlanır. İstemci tarafı:

```

1 <?php
2 $istemci=new SoapClient(null,array('uri'=>'http://ersindogan-
3 testuri','location'=>'http://localhost/test/sunucu.php'));
4 var_dump($istemci->gonder('mesaj','konu','kime'));
5 ?>

```

Yukarıdaki kodu anlatmadan önce, web sunucunuzda test diye bir klasör oluşturup içerisine sunucu.php ve istemci.php adında iki dosya ekleyin. sunucu.php dosyasına sunucu ile ilgili kodları ekleyin, istemci.php dosyasına da istemci ile ilgili kodları kopyalayıp dosyaları kaydedin. Daha sonra adres satırından http://localhost/test/istemci.php yazdığımızda

string 'mesaj gönderildi' (length=16)

biçiminde bir mesaj verecektir.

Çoğunlukla bu tür işlemleri aynı makinada, aynı platformda, hatta aynı programlama dilinde bile yapılamaz. Taraflar hep farklıdır, ama anlaşma metodu ortaktır. İşte bu ortak nokta yardımı ile uzak sistemden bir yordam çağrısı yaparak işlemlerimizi gerçekleştirebiliriz.

PHP ile Web Servisi Oluşturmak

Öncelikle Web Servisi Nedir?

Web Servisi kısaca başkalarının yazdığı uygulamalar ile kendi uygulama arasında bir köprü diyebiliriz. Örneği tersine de çevirebiliriz. Buradaki en önemli nokta platform bağımsızlığıdır. PHP ile yazdığınız bir servis .NET ile haberleşip data alışverişinde bulunabilir. Veya kullanıcı işlemleri bu yöntemle gerçekleştirilebilir.

PHP ile web servisi yazabilmek veya başka servislerle haberleşebilmek için 4 ayrı kütüphane mevcuttur. Bunlar OAuth, SCA, SOAP, XML-RPC.

SOAP (Simple Object Access Protocol – Basit Nesne Erişim Protokolü), bilgisayarlar ve networklar arasında veri değişimi olanağı sağlayan xml temelli standarttır.

Avantajları

Platform ve dil bağımsızlığı(XML’yi destekleyen bütün işletim sistemlerinde ve yazılım dillerinde mesaj gönderimi ve iletimi sorunsuz olarak gerçekleşir.)

Standart HTTP protokollerini kullanır(her hangi bir ayrı port açmaya ihtiyaç yoktur) Önceden tanımlanmış veri değişim şeması çerçevesinden veri gönderimi alımı olanağını sağlar.

soap kütüphanesini kurmak için:

- 1 sudo apt-get install php-soap
- 2 sudo service apache2 restart

komutları ile soap’ı kurup apache’yi restart ediyoruz.

Projemiz klasik olarak bir kitap satış sitesinin diğer kitap satış sitelerine vereceği arama ve kitap detay servisi olacak.

PHP Soap kütüphanesinde SOAPServer, SOAPClient, SOAPFault, SOAPHeader, SOAPParams, SOAPVar sınıfları bulunuyor. Bizim esaslı bir şekilde kullanacağımız sınıflar SoapServer, SoapClient ve SoapFault olacak.

Hemen SoapServer Sınıfını inceleyerek işe başlayalım. İşte SoapServer sınıfının yöntemleri:

SoapServer:: addFunction : Yazdığımız yöntemleri servise ekler. SOAP_FUNCTIONS_ALL ile bütün metotları ekleyebiliyoruz.

SoapServer:: __construct : SoapServer Sınıfı kurucu yöntemi.

SoapServer:: getFunctions : Tanımlı yöntemlerin listesini döndürür.

SoapServer:: handle : Web Servisinin isteklere hazır halde bekletir

SoapServer:: setClass : addFunction yönteminin benzeridir.

SOAPClient:

SoapClient::__call — soap sunucusundan tanımlı bir metota çağrı yapar. aldığı iki değer vardır birincisi metot ismi ikincisi gönderilecek değerler.

SoapClient::__getFunctions — Kullanılabilir metotların listesi

SoapClient::__getLastRequest — Son isteklerin döndürür.

SoapClient::__getLastResponse — Son gönderilen cevaplar.

ilk olarak kitap arama ve kitap detayları için oluşturulan sınıf:

```
1 class Book{
2
3   function BookTitle($BookID){
4     $sql=mysql_query(&quot;select * from Book where BookID=$BookID&quot;);
5     if(mysql_num_rows($sql)&lt;1){
6       return &quot;böyle bir kitap bulunamadı&quot;;
7     }
8     else {
9       $row=mysql_fetch_assoc($sql);
10      return $row;
11    }
12  }
13
14
15  function BookSearch($keywords){
16    $sql=mysql_query(&quot;select * from Book where title LIKE
17      '%$keywords%'&quot;);
18    if(mysql_num_rows($sql)&lt;1){
19      return &quot;Arama sonucunda kitap bulunamadı.&quot;;
20    }
21    else {
22      if(mysql_num_rows($sql)&gt;1){
23        while($row=mysql_fetch_assoc($sql)):
24          $result[]=$row;
25        endwhile;
26      }
27    }
28  }
```

```
27     $result=mysql_fetch_assoc($sql);
28
29     }
30     return $result;
31     }
32 }
33 }
```

Şimdi Soap servisimizi yazıp istekler için ;

```
1 $SoapServer=new SoapServer(NULL, array('uri' => "http://localhost/"));
2 $SoapServer->setClass('Book');
3 $SoapServer->handle();
```

Artık bir Client oluşturup istekte bulunma:

```
1 $client=new SoapClient(NULL, array('location' =>
    "http://localhost/BookClass.php", 'uri' => "http://localhost/"))
2 print_r($client->BookSearch('deneme kitabı'));
```

Db bağlantısı, arama keywordlerinin sql injection kontrollerinin yapılması gibi pek çok güvenlik kontrolüne ihtiyaç duyuyor.

PHP ile SOAP Web Servisi Oluşturmak

PHP ile basit bir SOAP web servisi nasıl oluşturulur?

SOAP Nedir ve Ne İşe Yarar?

SOAP (Simple Object Access Protocol – Basit Nesne Erişim Protokolü) kısaca farklı platformları ve programları haberleştirmek için kullanılan **XML** alt yapısıyla haberleşen bir web servisi.

Nasıl Çalışır?

XML dilini kullanarak karşı sunucuyla iletişim kurarak çalışmaktadır. Karşı sunucudan yapacağınız metod çağırımını ve o metoda göndereceğiniz parametreleri XML diliyle SOAP standardında yazıp karşı sunucuya **raw data POST** ederek çalışır.

SOAP Server Nedir ?

PHP 'de SOAP server, gelen SOAP isteklerini karşılayıp belirtilen fonksiyon yada metodlara yönlendiren bir dosyadır.

WSDL Nedir ?

WSDL (Web Services Description Language – Web Servisleri Tanımlama Dili), müşteri/kullanıcılarınızın sisteminiz üzerinden veri çekerken hangi metodları ve hangi parametreleri kullanacağını, hangi parametrelerin zorunlu olduğunu ve parametre tiplerinin ne olduğunu içeren dosyadır. Kısaca müşteri/kullanıcılarınızın için hareket tanımıdır.

Kısıtlamalar ?

- PHP5 ve üzeri versiyon kullanıyor olmak
- PHP ayarlarında (php.ini) SOAP uzantısının (extention) etkinleştirilmiş olması

SOAP Server İçin Lokasyon Önerisi?

Proje Dosyaları :

lib/

lib/Products.class.php

lib/nusoap.php

mod/

mod/client.php

mod/client_debug.php

client_getAllBooks.php

mod/create_wSDL.php rpc/

rpc/products.wSDL

rpc/productsService.php

Ürünlerimizi web servis aracılığıyla müşteri/kullanıcılarımıza gönderen bir yapı üzerinden gidilecektir. Mevcut yapınızdaki sınıf ya da fonksiyonları kullanabileceğiniz gibi mevcut değilse örneğimiz üzerinden ya da ihtiyaçlarınız doğrultusunda sınıf oluşturulabilir.

Ürün sınıfımızı yakından tanıyalım:

Exception sınıfını genişletir ve bir çok metodunu devralır. Özelliği ise Soap Server 'ın bu istisnayı yakalayıp istemciye direkt olarak bu mesajı dönmesidir.

Dosya Adı: **Products.class.php**

Class Adı: **Products**

Dosya Yolu: **lib/**

Metodların İmzaları :

- array **getAllNotebooks()**
Hiç bir parametre almaz ve tüm notebookları array olarak döndürür.
- string **checkNotebookStatus(integer \$notebookCode)**
Notebook kodunu parametre olarak alır ve notebook stok durumunu içeren string döndürür
(VAR/YOK).
- array **getAllBooks()**
Hiç bir parametre almaz ve tüm kitapları array olarak döndürür.
- string **checkBookStatus(string \$ISBN)**
Kitap ISBN numarasını parametre olarak alır ve kitap stok durumunu içeren string döndürür
(VAR/YOK).

Products.class.php dosyası içerisinde stok durumu dönen metodlarda Soap exception örneği için bilerek birer ürün eksik girilmiştir.

Products.class.php dosyasından bir kesit:

```
1
2
3
4 <?php
5
6 /**
7  * Urunlerin listesi, urun detayi ve stok durum kontrolu yapar.
8  *
9  * @author mustafa.kirimli
10  * @version 1.0
11  */
12 class Products {
13
14     /**
15      * Tum notebooklari dondurur
16      * @return array notebook listesi
17      */
18     public function getAllNotebooks() {
19
20         $notebooks = array();
21         $notebooks[] = array("code" => "123456", "brand" => "ASUS");
22         $notebooks[] = array("code" => "123457", "brand" => "LENOVO");
23         $notebooks[] = array("code" => "123458", "brand" => "TOSHIBA");
24
25         return $notebooks;
26     }
27
28     /**
29      * Notebook stok kontrolu yapar.
30      * @param integer $notebookCode notebook code
31      * @return string notebook urun stok durumu
32      */
33     public function checkNotebookStatus($notebookCode) {
34         $nbStatus = array("123456" => "VAR", "123458" => "YOK");
35         if (array_key_exists($notebookCode, $nbStatus)) {
36             return $nbStatus[$notebookCode];
37         } else {
38             throw new SoapFault("Notebook bulunamadı!", 8);
39         }
40     }
41 }
42 ..
43
44
45
46
47
48
```

WSDL Create İşlemi

WSDL oluşturmak için nusoap kütüphanesini kullanılacaktır.

create_wsd.php dosyası :

```
1
2  <?php
3
4  // wsd cache 'ini devre disi birak
5  ini_set("soap.wsd_cache_enabled", "0");
6
7  // nusoap dosyasini dahil et
8  include '../lib/nusoap.php';
9
10 // soap_server nesnesi olustur
11 $server = new soap_server();
12
13 // WSDL in ilk parametrelerini ver
14 $server->configureWSDL('ProductsService', // Servis adi
15                        'urn:ProductsService', // namespace
16                        'http://ws.mustafakirimli.com/rpc/productsService.php' // Servis
17                        adresi
18                        );
19
20 ..
```

Gerek wsdl oluştururken gerekse SoapServer için geliştirme yaparken en önemli noktanın yukarıdaki kodda bulunan ini_set bölümündeki **soap.wsdl_cache_enabled** olduğudur. Bunu ayarlamadığınızda yada bir şekilde (varsayılan olarak yada elle) **true/1** olduğunda wsdl dosyasını yine php ayarı olan **soap.wsdl_cache_ttl** 'de ayarlandığı kadar cache den okur. Dolayısıyla wsdl üzerinde yapılan değişiklikler test clientınıza çok geç yansiyabilir.

Çünkü her istekte bulunduğunuzda wsdl dosyasının karşı sunucudan alınıp indirilmesi çok maliyetli olabilir. Hele de çok fazla kullanılan bir SoapClient ise. include ile başlayan satırda ise **nusoap.php** dosyamızın bulunduğu yeri gösterip dahil edilir.

Sonraki satırda **soap_server** nesnesi oluşturulur.

Sonraki satırda ise (**\$server->configureWSDL**) wsdl için ilk atamaları yapılır. configureWSDL metodunun ilk üç parametresi ayarlanır. İlk parametre olan **\$serviceName** için web servisinin adını yazılır. Biz bu parametre için **ProductsService** i kullanılır. İkinci parametre olan **\$namespace** için ise; **urn:ProductsService** kullandık (Burada ProductsService yazan yere servis adı yazılır).

Üçüncü parametre (**\$endpoint**) içinde **http://ws.xxxxx.com/rpc/productsService.php** adresini kullanılır. İlk parametre hariç diğerleri isteğe bağlıdır. Bu aşamadan sonra notebooklar ve kitaplar için ayrı ayrı **ComplexType** tanımlamaları yapıp (wsdl 'e eklenmek üzere) metodlarımızı register metodu ile web servisi wsdl 'imize kaydedilir.

Neden ComplexType Tanımlanır ?

ComplexType tanımlamanın sebebi PHP den yada diğer bir çok dilden bilindiği gibi metotlara tamamı array yada bazıları array parametreler gönderilir. SOAP web servisinde array ile data almak (client 'dan) yada göndermek (server 'dan) istediğinizde **ComplexType** kullanılır. Ve web servisi için tanımladığımız metoda input yada output parametresinde **“tns:”** ile erişim için oluşturulan tipi kullanılır.

Örnek ComplexType Tanımı

```
1  <?php
2
3  // Notebook tipi tanımla
4  $server->wsdl->addComplexType(
5      'Notebook', // özel tip adi
6      'complexType', // tip
7      'struct', // compositor
8      'all', // restrictionBase
9      '', // elements
10     array(
11         // 'ozellikAdi' => array('name' => 'ozellikAdi',
12         //                                     'type' => 'ozellikTipi',
13         //                                     'minOccurs' => enazKacDefaOlacak,
14         //                                     'maxOccurs' =>
15         enazFazlaKacDefaOlabilir)
16         'code' => array('name' => 'code', 'type' => 'xsd:string'),
17         'brand' => array('name' => 'brand', 'type' => 'xsd:string')
18     )
19 )
```

```
16 );  
17  
18 ..  
19  
20
```

Metod 'un Array Tipinde Sonuç Dönmesi İçin Gereken ComplexType

```
1  
2 <?php  
3  
4 // NotebookList tipi tanımla  
5 $server->wsdl->addComplexType(  
6     'NotebookList', // özel tip adı  
7     'complexType', // tip 'struct', //  
8     compositor  
9     'all', // restrictionBase  
10    '', // elements  
11    array(  
12        // 'ozellikAdi' => array('name' => 'ozellikAdi',  
13        // 'type' => 'ozellikTipi',  
14        // 'minOccurs' => enazKacDefaOlacak,  
15        // 'maxOccurs' =>  
16        enazFazlaKacDefaOlabilir)  
17        'Notebook' => array('name' => 'Notebook',  
18        'type' => 'tns:Notebook',  
19        'minOccurs' => '0',  
20        'maxOccurs' => 'unbounded')  
21    )  
    );
```

Kullanıldığı yer aşağıdaki (**getAllNotebooks**) metodunun dönüş tipi parametresidir (**NotebookList**).

Yani döndüreceği değerleri belirler.

Tanım parametrelerinde tns ile bir önceki Notebook ComplexType 'ına göndermede bulunur.

Yani kısaca bu “**ComplexType**” tanımı/nesnesi **getAllNotebooks** metodunun **Notebook** tipinde birden çok değer dönebileceği (**maxOccurs = unbounded**) anlamına gelmektedir.

Web Servisi İçin Metod Tanımlama

Web servisi için metod tanımlamayı register metodu ile yapılır. Söz dizimi biraz karışık gibi görünse de PHP ve benzer (syntax olarak) dillerden çokta farklı değildir. Metodun adını verirsiniz, giriş parametreleri ve çıkış parametrelerini verirsiniz. Bir kaç ek özelliklikle birlikte SOAP ta da aynen böyle yapılır.

Örnek Web Servisi Metodu

```
1
2  <?php
3
4  //  getAllNotebooks metodunu web servisi için tanımla $server-
5  >register('getAllNotebooks', // metod adı
6          array(), // giriş parametreleri
7          array('return' => 'tns:NotebookList'), // çıkış parametreleri
8          'urn:ProductsService', // namespace
9          'urn:ProductsService#getAllNotebooks', // soapaction 'rpc', // style
10         'encoded', // use
11         'Tum notebooklari dondurur' // dokuman
12 );
13
14 ..
```

Yukarıdaki örnekte **getAllNotebooks** metodunu tanımlanmıştır. Metod parametre almadığı için boş bir array yazılmış durumda. Ayrıca metod **NotebookList** (dosyada ComplexType olarak tanımlanmış hali mevcuttur) tipinde (**tns** ile yazarak yerleşik tip olduğunu söyleriz) bir parametre döndüreceğini tanımlamıştır..

WSDL Tanımını Oluşturmak ve Dosyaya Yazmak

```
1  <?php
2
3  // Servisi baslat
4  $HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ? $HTTP_RAW_POST_DATA : '';
5  $server->service($HTTP_RAW_POST_DATA);
6
7  // parametre olarak print geldiyse WSDL dosyasi olustur
8  if ($_GET["print"] == true) {
9      // bu sayfayi WSDL dosyasini gosterecek bicimde ac ve ciktisini al
10     $data =
11     file_get_contents("http://ws.mustafakirimli.com/mod/create_wsd1.php?wsdl");
12     // alinan ciktiyi wsd dosyasina yaz
13     file_put_contents("../rpc/products.wsdl", $data);
14 }
15 ..
```

Yukarıdaki kod ise servisi (sadece wsdl oluşturmak için) başlatır. Bu dosyayı tarayıcınızdan çalıştırırsanız WSDL i ekrana basacaktır. İsterseniz çıkan WSDL tanımını kaynağını görüntüle diyerek wsdl dosyanıza kaydedebilirsiniz.

Ayrıca yukarıdaki dosyayı direkt olarak çalıştırırsanız WSDL özetini ve metodlar hakkında bilgiler veren ekranları görebilirsiniz.

Metod Listesi Ekranı | Metod Detayları Ekranı

Bu dosya (create_wsd1.php) aslında bir web servisi olarak çalışabilir ancak biz bu dosyayı sadece WSDL dosyasını oluşturmak için kullanılır. Çünkü servis olarak **PHP5 'in yerleşik SOAP Server özelliğini** (ram 'den tasarruf için) kullanır.

Dosyanın en altındaki kısım ise (yukarıda da mevcut) dosyayı wsdl 'i gösterecek bir biçimde açıp wsdl dosyasına yazmayı sizin yerinize yapar.

SOAP Server Dosyamız

```
1  <?php
2
3  // wsdl cache 'ini devre disi birak
4  ini_set("soap.wsdl_cache_enabled", "0");
5
6  // web servisi icin kullanacagimiz metodlari iceren sinifi dahil et
```

```
6   require_once '../lib/Products.class.php';
7
8   // Soap Server nesnesi olustur
9   $soapServer = new SoapServer("products.wsdl", array('encoding' => 'UTF-
10  8'));
11
12  // Soap Server 'a Products sinifini kullanmasini soyle
13  $soapServer->setClass("Products");
14
15  // Soap Server 'i baslat ve gelen istekleri Products sinifina gonder
16  $soapServer->handle();
17  ?>
```

Dosyamız wsdl cach 'ini devre dışı bırakarak başlıyor (Bu ayarıda product/live ortamınızda 1 olarak ayarlayabilirsiniz). Daha sonra web servisimizin kullanacağı metodları içeren sınıfı dahil ediyoruz.

PHP5 ile yerleşik olarak gelen **SoapServer** nesnemizi oluşturup parametre olarak oluşturulan WSDL dosyasını veriyoruz (varsa kendi WSDL dosyanızı da verebilirsiniz). Son olarak setClass ile sınıfımızı SoapServer 'a kaydettirdikten sonra **handle** diyerek servisimizi başlatıyoruz.

WEB SERVICE ÇEŞİTLERİ

Temel olarak 2 çeşit Web Service vardır. Rest mimarisi yani **Restful** ve **SOAP** Web Servisleri vardır. İki arasında ki en derin farkı Restful ile xml, json, metin, html gibi istediğiniz türde çıktılar alıp gönderebilirken, SOAP Web Service kavramında sadece xml veriler ile işlem yapabilirsiniz.

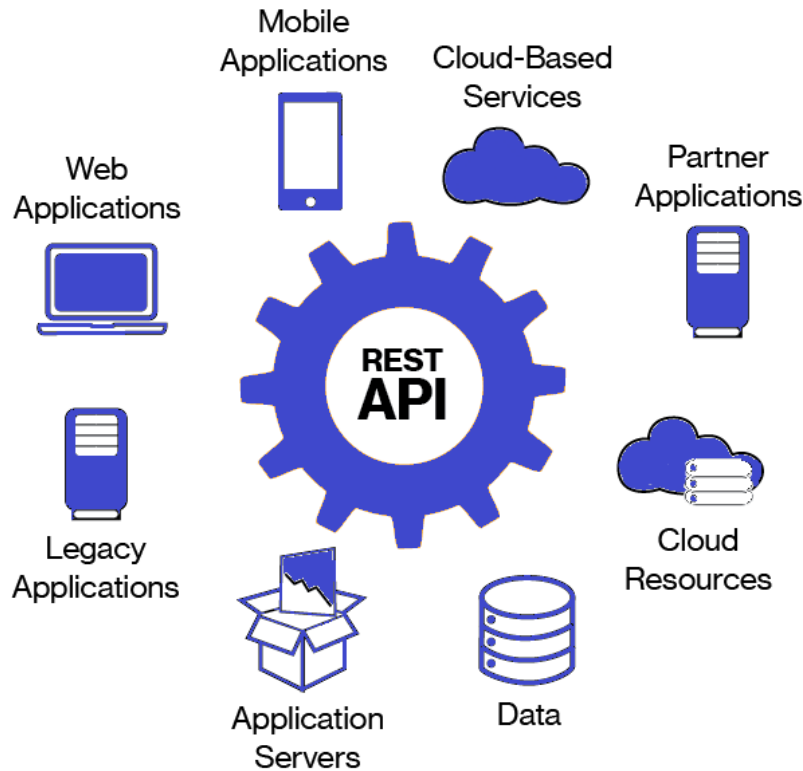
REST (REpresentational State Transfer)

REST istemci-sunucu arasında hızlı ve kolay şekilde iletişim kurulmasını sağlayan bir servis yapısıdır. Açılımı Representational State Transfer olan bu ifadeyi Türkçe'ye Temsili Durum Transferi diye çevirebiliriz.

REST, servis yönelimli mimari üzerine oluşturulan yazılımlarda kullanılan bir veri transfer yöntemidir. HTTP üzerinde çalışır ve diğer alternatiflere göre daha basittir, minimum içerikle veri alıp gönderdiği için de daha hızlıdır. İstemci ve sunucu arasında XML veya JSON verilerini taşıyarak uygulamaların haberleşmesini sağlar. REST standartlarına uygun yazılan

web servislerine **RESTful** servisler denir. RESTful servisler SOAP, RPC'nin aksine basit ve hafiftirler.

Basit olmalarının yanında oldukça da esnek ve yeteneklidirler. Aslında tipik Web Servislerle yapılabilecek herşey RESTful servislerle yapılabilir. Ayrıca mimari olarak nasıl olması, ne gibi özelliklere sahip olması hakkında belli yönergeler olsa da, burada SOAP gibi keskin standartları olan bir mimari yoktur. Üzerine çoğu platformda (C#,JAVA vs.), bir sürü Framework yazılmış durumda, fakat birçok platformun standart library'leri kullanılarak, hızlıca RESTful Servisler geliştirilebilir.



REST stateless'dır, yani durum bilgisini saklamaz. Biraz daha açalım; REST standartlarında istemci-sunucu arasında taşınan verilerde ekstra başlık bilgileri saklanmaz, istemciye ait detaylar bulunmaz, bu bilgiler istemci-sunucu arasında taşınmaz. Dolayısıyla servis yönelimli uygulamalarda REST bize lightweight bir çözüm yapısı sunar.

RESTful servisler veri iletiminde farklı HTTP metotlarını kullanmaktadır. Bunlar GET, POST, PUT, DELETE metotlarıdır. Yapılan HTTP request'i için çağrılan URL ile beraber HTTP method bilgisi bahsi geçen 4 metottan biri olarak seçilir ve sunucu yapılan talebin kayıt üzerine nasıl etki edeceğini buna göre belirler. Böylece proxy ihtiyacı ortadan kalkmış olur ve

platform bağımsız yapılar kurmak kolaylaşır. Şu anki modern uygulamalarda bu Method'ları harfiyen kullanmak bir zorunluluk olmasa da, standartlara uymak, işlem tutarlılığı ve güvenliği açısından önemlidir.

REST Request Tipleri:

GET : Belirtilen collection'ın URI'lerini veya detaylarını listelemede kullanılır.

PUT: Bütün bir collection'ı başka bir collection'la yer değiştirmek için kullanılır.

POST: Yeni bir collection oluşturmak için kullanılır ve yeni oluşturulan collection'ın URI'si döndürülür.

DELETE: Tüm Collection'ı silmek için kullanılır.

REST İLE SOAP FARKI

Güvenlik: SOAP üzerinde güvenlik(security) sağlamak daha kolay ve hızlı şekilde olabiliyorken, bu durum REST için karmaşık olabiliyor. REST için güvenlik söz konusu olduğunda SOAP'a göre biraz zayıf kalmaktadır.

Data Yapısı: REST ile JSON, XML hatta TEXT ile çalışabiliyorken, SOAP ile XML kullanmalıyız. REST bu sayede daha kullanışlı olabiliyor. JSON ile daha küçük boyutlarda verilerle işlemler gerçekleştirebiliyorsunuz. Eğer data boyutları uygulamanız için önem arz ediyorsa REST kullanmak sizin için uygun olacaktır.

Uygulama Hızı: Uygulamanızın daha hızlı çalışmasını istiyorsanız REST kullanmanız yararınıza olacaktır.

Destek: SOAP için geliştirici araçları daha iyidir. Kullanmamız için bize yardımcı olacak daha çok kaynak bulabiliriz. REST'in dökümantasyonu SOAP'a göre daha azdır.

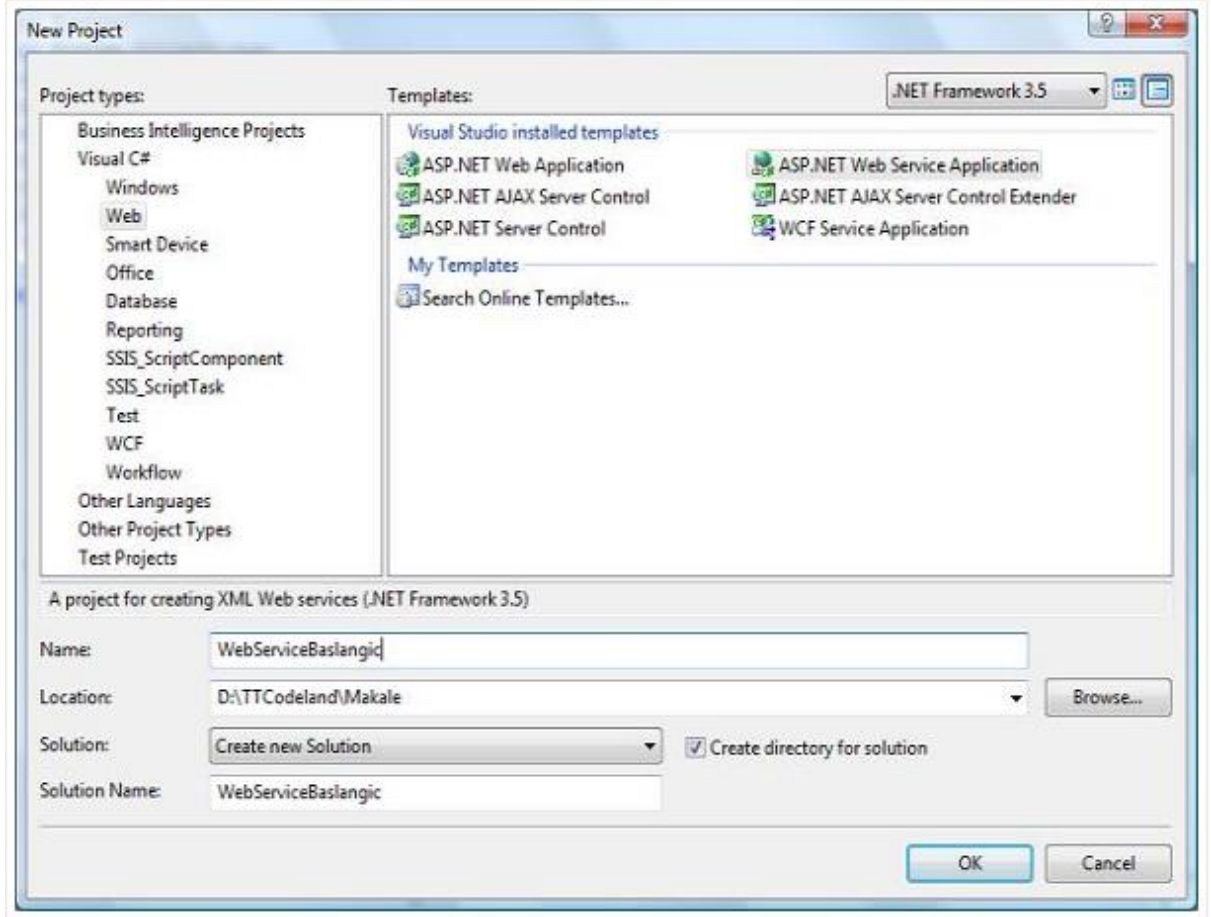
REST yaklaşımı HTTP metotlarını kullanarak işlerini görür. GET, POST, PUT, DELETE vs.

SOAP servisleri RPC(Remote Process Call yani uzaktaki bir prosedürün çağırılması) çalışma yöntemini kullanır, WS-* gibi güvenlik protokollerini içerisinde barındırır, state bilgisini request ve response'larda saklar.

REST mimarisi SOAP'a göre oldukça esnektir ve hafiftir, taşınan veri miktarı daha azdır, entegrasyonu da daha kolaydır.

ÖRNEK XML WEB SERVİS UYGULAMASI

Aldığı parametreye göre veri tabanından sorgu yaparak sonucu geri döndüren basit bir web servis metodu oluşturacağız. Visual Studio 2008 de New Project -> Visual C# -> Web menusu altından ASP.NET Web Service Application seçeneğini seçerek devam edelim.



Yukarıdaki ekranda Ok butonuna bastığınız zaman web servisiniz otomatik olarak oluşturulacak. Visual Studio web servisimizin çalışması için gerekli aşağıdaki kodları oluşturacaktır.

```

namespace WebServiceBaslangic
{
    /// <summary>
    /// Summary description for Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [ToolboxItem(false)]
    // To allow this Web Service to be called from script, using ASP.NET
    AJAX, uncomment the following line.
    // [System.Web.Script.Services.ScriptService]
    public class Service1 : System.Web.Services.WebService
    {

        [WebMethod]
        public string HelloWorld()
        {
            return "Hello World";
        }
    }
}

```

Yukarıdaki kodu inceleyecek olursak HelloWorld adında bir metod oluşturulmuş ve çağırıldığı durumda “Hello World” string değerini döndürmekle görevli. Burada dikkat edilmesi gereken [WebMethod] kısmı. Eğer web servisimize yazdığımız fonksiyona dışarıdan erişilmesini istiyorsanız [WebMethod] etiketini metod isminin hemen üstüne koymak zorundasınız.

Web metodu aldığı parametre ile veri tabanında sorgu çalıştırarak veri tabanından dönen verileri, metodu çağırان istemciye gönderecektir. Bunun için ilk olarak kullanacağımız veri tabanı tablomuzu oluşturalım. Ben örneğimizde Kişi adında bir tablo kullanacağım. Tablomuzun scripti aşağıdaki gibidir.

```

mysql> create table Kisi (No int not null, Ad varchar(45), Soyad varchar(45),
TcKimlikNo int, Numara int, Cinsiyet char, DogumYeri varchar(45));
Query OK, 0 rows affected (0.22 sec)

mysql>

```

Veri eklediğimizde tablomuzun içeriği aşağıdaki gibidir.

```

mysql> select * from Kisi ;
+-----+-----+-----+-----+-----+-----+-----+
| No   | Ad    | Soyad | TcKimlikNo | Numara | Cinsiyet | DogumYeri |
+-----+-----+-----+-----+-----+-----+-----+
| 2    | Metin | Tosun | 76543244233 | 124    | E        | Ankara    |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

Web servisimizi kod kısmını açarak aşağıdaki kodları ekleyin.

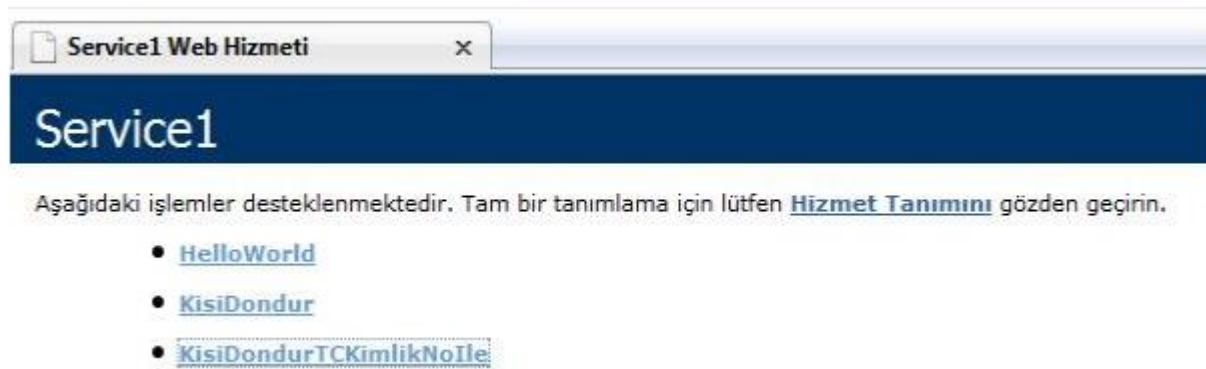
```

[WebMethod]
public DataSet KisiDondur(string ad)
{
    string query = String.Format("SELECT * FROM Kisi WHERE Ad = '{0}'",ad);
    return SorguCalistir(query);
}

[WebMethod]
public DataSet KisiDondurTCKimlikNoIle(string tcKimlikNo)
{
    string query = String.Format("SELECT * FROM Kisi WHERE TCKimlikNo = '{0}'", tcKimlikNo);
    return SorguCalistir(query);
}

```

Yukarıda gördüğümüz gibi KisiBilgilerini döndüren iki fonksiyon tanımladık ve tanımlamalarının hemen üstüne [WebMethod] ekledik. Bağlantı ayarlarını da yaptıktan sonra Web Servisimizi çalıştırarak sonuçlara bakalım :



Yukarıda görüldüğü gibi [WebMethod] eklediğimiz tüm fonksiyonlarımız listelendi. Bunlardan KisiDondur linkini tıklayarak devam edelim.

Service1 Web Hizmeti

Service1

İşlemlerin tam listesi için [burayı](#) tıklatın.

KisiDondur

Sinama

HTTP POST protokolünü kullanarak işlemi sınamak için, 'Çağır' düğmesini tıklatın.

Parametre	Değer
ad:	<input type="text" value="Metin"/>

Çağır

SOAP 1.1

Aşağıdaki örnekte bir SOAP 1.1 isteği ve yanıtı yer alıyor. Gösterilen yer tutucular gerçek değerlerle değ

```
POST /Service1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://yasilimutfagi.com/KisiDondur"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <KisiDondur xmlns="http://yasilimutfagi.com/">
      <ad>string</ad>
    </KisiDondur>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

Ad kısmına Metin yazarak Çağır dediğimizde Metin adlı bir kişi tablomuzda var ise Metine ait tüm bilgiler aşağıda görüldüğü gibi XML formatında bize dönecektir.

```

<DataSet>
-
  <xs:schema id="NewDataSet">
    -
      <xs:element name="NewDataSet" msdata:IsDataSet="true"
msdata:UseCurrentLocale="true">
        -
          <xs:complexType>
            -
              <xs:choice minOccurs="0" maxOccurs="unbounded">
                -
                  <xs:element name="Table">
                    -
                      <xs:complexType>
                        -
                          <xs:sequence>
                            <xs:element name="No" type="xs:int" minOccurs="0"/>
                            <xs:element name="Ad" type="xs:string" minOccurs="0"/>
                            <xs:element name="Soyad" type="xs:string" minOccurs="0"/>
                            <xs:element name="TCKimlikNo" type="xs:string"
minOccurs="0"/>
                            <xs:element name="Numara" type="xs:int" minOccurs="0"/>
                            <xs:element name="Cinsiyet" type="xs:string" minOccurs="0"/>
                            <xs:element name="DogumYeri" type="xs:string"
minOccurs="0"/>
                          </xs:sequence>
                        </xs:complexType>
                      </xs:element>
                    </xs:choice>
                  </xs:complexType>
                </xs:element>
              </xs:schema>
            -
              <diffgr:diffgram>
                -
                  <NewDataSet>
                    -
                      <Table diffgr:id="Table1" msdata:rowOrder="0">
                        <No>2</No>
                        <Ad>Metin</Ad>
                        <Soyad>Tosun</Soyad>
                        <TCKimlikNo>76543244233</TCKimlikNo>
                        <Numara>124</Numara>
                        <Cinsiyet>E</Cinsiyet>
                        <DogumYeri>Ankara</DogumYeri>
                      </Table>
                    </NewDataSet>
                  </diffgr:diffgram>
                </DataSet>
              </diffgr:diffgram>
            -
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:schema>
  </DataSet>

```

RAPORDA YAPILMASI İSTENENLER

-Web servis yazmayı anladıysanız “eOkul” isimli **bir MYSQL veritabanı** oluşturunuz.

- eOgrenci ve eBolum tablolarını bu VTYS için yaratınız.

- eOgrenci (OgrenciNo, OgrenciAd, OgrenciSoyad, SınıfNo, DanışmanAdı, BolumID)
- eBolum (BolumID, BolumAd)

-Bu tablodan OgrenciNo bilgisi gönderilen öğrencinin adını, soyadını, sınıfını ve danışman adını döndüren Soap servis veya Restfull servis yazınız. Yazmış olduğunuz web servisine bir

asp.net sayfasından veya bir jsp sayfadan erişerek ürünId bilgisini vererek ürün özelliklerini listeleyiniz.

NOT : - Bu föy kapsamında istenilen uygulamayı seçtiğiniz bir yazılım teknolojisi (Java, PHP veya .NET) kullanarak her bir web servis geliştirme yöntemi için (SOAP teknolojisi kullanılarak ve Restfull Servis kullanarak) ayrı ayrı gerçekleştiriniz.

- Aralarında ki farklılıkları açıklamaları ve karşılaştırmalı olarak raporunuzda anlatınız.