

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



Veri Tabanı Yönetimi Laboratuvarı

Föy 5

15060007 Kübra Nur Saruhan

Föyde istenilen eOgrenci ve eBolum tabloları oluşturuldu ve Insert into ile 2 veri eklendi.

```
1 • use eOkul;
2 • CREATE TABLE eBolum(
3     bolumId int NOT NULL PRIMARY KEY,
4     bolumAd varchar(255)
5 );
6 • CREATE TABLE eOgrenci(
7     ogrenciNo int NOT NULL PRIMARY KEY ,
8     ogrenciAd varchar(255),
9     ogrenciSoyad varchar(255),
10    sinifNo int,
11    danismanAdi varchar(255),
12    bolumId int,
13    FOREIGN KEY(bolumId) REFERENCES eBolum(bolumId)
14 );
15
```

Output

Action Output

#	Time	Action
✓ 1	17:51:43	use eOkul
✓ 2	17:51:43	CREATE TABLE eBolum(bolumId int NOT NULL PRIMARY KEY, bolumAd
✓ 3	17:51:44	CREATE TABLE eOgrenci(ogrenciNo int NOT NULL PRIMARY KEY , ogre

```
1 • use eOkul;
2 • INSERT INTO eBolum(bolumId, bolumAd
3     )VALUES(1, 'Bilgisayar Mühendisi'),
4     (2, 'İnşaat Mühendisi');
5 • INSERT INTO eOgrenci(
6     ogrenciNo, ogrenciAd, ogrenciSoyad, sinifNo, danismanAdi, bolumId
7     )VALUES(1, 'Kübra Nur', 'Saruhan', 1, 'danışman1', 1 ),
8     (2, 'Sümeyra', 'Aydın', 3, 'danışman3', 2)
9
10
```

Output









Action Output

#	Time	Action
✓ 1	17:58:24	use eOkul
✓ 2	17:58:24	INSERT INTO eBolum(bolumId, bolumAd)VALUES(1, 'Bilgisayar Mühendisi'). (2, 'İnşaat Mühendisi')
✓ 3	17:58:24	INSERT INTO eOgrenci(ogrenciNo, ogrenciAd, ogrenciSoyad, sinifNo, danismanAdi, bolumId)VALI

SOAP WEB SERVİSİ

KişiGetir adında fonksiyonu oluşturuldu. Parametresine id verildi. Girilen id mysql veri tabanından çekilen öğrenci numarası ile aynı olan öğrencinin bilgilerini xml olarak göstermesi sağlandı.

```
namespace WebService
{
    /// <summary>
    /// WebService1 için özet açıklama
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // Bu Web Hizmeti'nin, ASP.NET AJAX kullanılarak komut dosyasından çağrılmasına, aşağıdaki satırı açıklamadan kal
    // [System.Web.Script.Services.ScriptService]
    public class WebService1 : System.Web.Services.WebService
    {
        [WebMethod]
        public DataSet KisiGetir(int id)
        {
            MySqlConnection connection = new MySqlConnection();
            connection.ConnectionString = "server=localhost; uid=root; pwd=1331; database=eOkul; SslMode = none";
            connection.Open();
            MySqlDataAdapter adapter = new MySqlDataAdapter("Select ogrenciAd, ogrenciSoyad, sinifNo, " +
                "danismanAdi From eOgrenci Where ogrenciNo = '" + id + "'", connection);
            adapter.SelectCommand.ExecuteNonQuery();
            DataSet dt = new DataSet();
            adapter.Fill(dt);
            return dt;
        }
    }
}
```

     localhost:64769/WebService1.asmx   

WebService1

Aşağıdaki işlemler desteklenmektedir. Tam bir tanımlama için lütfen [Hizmet Tanımını](#) gözden geçirin.

- [KisiGetir](#)

Bu web hizmeti, varsayılan ad alanı olarak http://tempuri.org/ kullanıyor.

Öneri: Varsayılan ad alanını XML Web hizmeti genel kullanıma açılmadan önce değiştirin.

Her XML Web hizmetinin, istemci uygulamalarının kendisini Web üzerindeki diğer hizmetlerden ayırt edebilmesi için benzersiz bir ad alanı olmalıdır. Geliştirilmekte olan XML Web hizmetleri için htt daha kalıcı bir ad alanı kullanmalıdır.

XML Web hizmetiniz sizin denetiminizde olan bir ad alanı tarafından tanımlanmalıdır. Örneğin şirketinizin Internet etki alanı adını ad alanının bir parçası olarak kullanabilirsiniz. Pek çok XML Web h etmeleri gerekmez. (XML Web hizmeti ad alanları, URI'lerdir.)

ASP.NET kullanılarak oluşturulan XML Web hizmetlerinde, varsayılan ad alanını değiştirmek için WebService özniteliğinin Namespace özelliği kullanılabilir. WebService özniteliği, XML Web hizmeti n "http://microsoft.com/webservices/" olarak belirleyen bir kod örneği görülmüyor:

C#

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

Visual Basic

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

C++

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

XML ad alanları hakkında daha fazla bilgi için [XML'de ad alanları](#) ile ilgili W3C önerilerine bakın.

WSDL konusunda daha fazla bilgi için, [WSDL Belirtimi'ne](#) bakın.

URI'ler hakkında daha fazla bilgi için, [RFC 2396](#) belgesine bakın.

WebService1

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

KisiGetir

Sinama

HTTP POST protokolünü kullanarak işlemi sinamak için, 'Çağır' düğmesini tıklayın.

Parametre	Değer
id:	<input type="text"/>

SOAP 1.1

Aşağıdaki örnekte bir SOAP 1.1 isteği ve yanıtı yer alıyor. Gösterilen [yer tutucular](#) gerçek değerlerle değiştirilmelidir.

```
POST /WebService1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/KisiGetir"

<?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>
    <KisiGetir xmlns="http://tempuri.org/">
      <id>int</id>
    </KisiGetir>
  </soap:Body>
</soap:Envelope>

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

<?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>
    <KisiGetirResponse xmlns="http://tempuri.org/">
      <KisiGetirResult>
        <xsd:schema>schema</xsd:schema>xml</KisiGetirResult>
      </KisiGetirResponse>
    </soap:Body>
  </soap:Envelope>
```

id=1 yazıldığında öğrenci numarası 1 olarak belirttiğimiz öğrencinin föyde istenildiği gibi adı, soyadı, sınıfı ve danışman adı gösterildi.

```
<?xml version="1.0" encoding="UTF-8"?>
- <DataSet xmlns="http://tempuri.org/">
  - <xs:schema xmlns="" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:xs="http://www.w3.org/2001/XMLSchema" id="NewDataSet">
    - <xs:element msdata:UseCurrentLocale="true" msdata:IsDataSet="true" name="NewDataSet">
      - <xs:complexType>
        - <xs:choice maxOccurs="unbounded" minOccurs="0">
          - <xs:element name="Table">
            - <xs:complexType>
              - <xs:sequence>
                <xs:element name="ogrenciAd" minOccurs="0" type="xs:string"/>
                <xs:element name="ogrenciSoyad" minOccurs="0" type="xs:string"/>
                <xs:element name="sinifNo" minOccurs="0" type="xs:int"/>
                <xs:element name="danismanAdi" minOccurs="0" type="xs:string"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
- <diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
  - <NewDataSet xmlns="">
    - <Table diffgr:id="Table1" msdata:rowOrder="0">
      <ogrenciAd>Kübra Nur</ogrenciAd>
      <ogrenciSoyad>Saruhan</ogrenciSoyad>
      <sinifNo>1</sinifNo>
      <danismanAdi>danisman1</danismanAdi>
    </Table>
  </NewDataSet>
</diffgr:diffgram>
</DataSet>
```

RESTFULL WEB SERVİSİ

Restfull Web Servis C# anlatımı : <https://www.youtube.com/watch?v=i1HSG7ttDtM> adresinden faydalanılmıştır.

Eokul sınıfımızın get, set modelleri oluşturulmuştur.

```
Eokul.cs* EokulPersistence.cs EokulController.cs
WebServiceRest
1  using System;
2      using System.Collections.Generic;
3      using System.Linq;
4      using System.Web;
5
6  namespace WebServiceRest.Models
7  {
8      public class Eokul
9      {
10         public int OgrenciNo { get; set; }
11         public string OgrenciAd { get; set; }
12         public string OgrenciSoyad { get; set; }
13         public int SinifNo { get; set; }
14         public string DanismanAdi { get; set; }
15         public int BolumId { get; set; }
16     }
17 }
```

Restfull web servisi için gereken controllers oluşturuldu. Array List, GET metodunu kullanarak bütün öğrencilerin listelenmesi sağlandı. Oluşturulan eokulPersistence'den GetButunOgrenciler fonksiyonu çağırılmıştır. Fonksiyon sonraki sayfalarda anlatılmıştır. Eokul sınıfını kullanarak oluşturulan GET metodu ile gönderilen id ile sadece istenilen öğrenci bilgisi gösterilir.

```
EokulPersistence.cs EokulController.cs
WebServiceRest.Controllers
1  using System.Net.Http;
2  using System.Web.Http;
3  using WebServiceRest.Models;
4  using System.Collections;
5
6  namespace WebServiceRest.Controllers{
7      public class EokulController : ApiController{
8          // GET: api/Eokul
9          public ArrayList Get(){
10              EokulPersistence eokulPersistence = new EokulPersistence();
11              return eokulPersistence.GetButunOgrenciler();
12          }
13
14          // GET: api/Eokul/5
15          public Eokul Get(int id)
16          {
17              EokulPersistence eokulPersistence = new EokulPersistence();
18              Eokul eokul = eokulPersistence.GetEokul(id);
19              return eokul;
20          }
21
22          // POST: api/Eokul
23          public void Post([FromBody]string value)
24          {
25          }
26
27          // PUT: api/Eokul/5
28          public void Put(int id, [FromBody]string value)
29          {
30          }
31
32          // DELETE: api/Eokul/5
33          public void Delete(int id)
34          {
35          }
36      }
37 }
```

İlk olarak MySql veri tabanına bağlantısı sağlanmıştır. GetEokul fonksiyonuna öğrenci numarası gönderiler istenilen öğrenci bilgisi veri tabanından sql sorgu ile bulunmuştur. Eokul sınıfı kullanılarak veriler kaydedilmiş ve return ile döndürülmüştür.

```
namespace WebServiceRest{
    public class EokulPersistence{
        MySqlConnection connection = new MySqlConnection();
        public EokulPersistence(){
            connection.ConnectionString = "server=localhost; uid=root; pwd=1331; database=eOkul; SslMode = none";
            try{
                connection.Open();
            }
            catch (MySqlException ex){
            }
        }
        public Eokul GetEokul(int ogrenciNo){
            Eokul eokul = new Eokul();
            MySqlDataReader reader = null;
            MySqlCommand command = new MySqlCommand("Select * From eOgrenci Where ogrenciNo = " + ogrenciNo + "", connection);
            reader= command.ExecuteReader();
            if (reader.Read()){
                eokul.OgrenciNo = reader.GetInt32(0);
                eokul.OgrenciAd = reader.GetString(1);
                eokul.OgrenciSoyad = reader.GetString(2);
                eokul.SinifNo = reader.GetInt32(3);
                eokul.DanismanAdi = reader.GetString(4);
                eokul.BolumId = reader.GetInt32(5);
                return eokul;
            }
            else{
                return null;
            }
        }
    }
}
```

localhost:52422/api/Eokul/1; buradaki 1 istenilen öğrencini numarasıdır.



localhost:52422/api/Eokul/1

```
{"OgrenciNo":1,"OgrenciAd":"Kübra Nur","OgrenciSoyad":"Saruhan","SinifNo":1,"DanismanAdi":"danışman1","BolumId":1}
```

2 olarak girildiğinde farklı öğrenci bilgileri listelenmektedir.



localhost:52422/api/Eokul/2

```
{"OgrenciNo":2,"OgrenciAd":"Sümeyra","OgrenciSoyad":"Aydın","SinifNo":3,"DanismanAdi":"danışman3","BolumId":2}
```

Array Liste veri tabanından gelen bütün öğrencileri bilgileri kaydedilip return ile döndürülmüştür.

```
public ArrayList GetButunOgrenciler()
{
    ArrayList eokulArray = new ArrayList();
    MySqlDataReader reader = null;
    MySqlCommand command = new MySqlCommand("Select * From eOgrenci", connection);
    reader = command.ExecuteReader();
    while (reader.Read())
    {
        Eokul eokul = new Eokul();
        eokul.OgrenciNo = reader.GetInt32(0);
        eokul.OgrenciAd = reader.GetString(1);
        eokul.OgrenciSoyad = reader.GetString(2);
        eokul.SinifNo = reader.GetInt32(3);
        eokul.DanismanAdi = reader.GetString(4);
        eokul.BolumId = reader.GetInt32(5);
        eokulArray.Add(eokul);
    }
    return eokulArray;
}
```

Localhost:52422/api/Eokul ile veri tabanındaki bütün öğrenciler listelenmiştir.

← → ↺ ⓘ localhost:52422/api/Eokul

```
[{"OgrenciNo":1,"OgrenciAd":"Kübra Nur","OgrenciSoyad":"Saruhan","SinifNo":1,"DanismanAdi":"danisman1","BolumId":1},
{"OgrenciNo":2,"OgrenciAd":"Sümeyra","OgrenciSoyad":"Aydın","SinifNo":3,"DanismanAdi":"danisman3","BolumId":2}]
```

Soap ve Restfull Servisleri Arasındaki Farklar

Rest standartlarına uygun yazılan web servislerine Restfull servisler denir.

- SOAP XML veri tipini desteklerken REST istenen veri türüyle işlem yapabilir. JSON veri tipi ile XML'den çok daha düşük boyutlarla veri tutulabildiği için REST ile daha hızlı işlem yapılabilir.
- SOAP için birçok geliştirme aracı mevcuttur, REST için geliştirme araçlarına ihtiyaç duyulmaz, tasarlaması kolaydır.
- SOAP; XML-Scheme kullanırken REST; URI-scheme kullanır yani metotlar için URI'ler tanımlanır.
- Her ikisi de HTTP protokolünü kullanırlar. Fakat REST için HTTP zorunluluğu varken SOAP; TCP, SMTP gibi başka protokollerle de çalışabilir.
- Test ve hata ayıklama aşaması REST için daha kolaydır. Çünkü HTTP hatalarını döndürür ve bunlar bir toola ihtiyaç duyulmadan görülebilir. SOAP için hata ayıklama araçları gerekebilir.
- REST basit HTTP GET metodunu kullandığı için cacheleme işlemi daha kolaydır. SOAP ile cacheleme yapabilmek için karmaşık XML requestleri yapılmalıdır.
- Güvenlik açısından SOAP daha gelişmiştir çünkü hazır yapılar bulunmaktadır.
- Dokümantasyon bakımından SOAP daha gelişmiştir ve daha fazla kaynak bulunmaktadır.