

# ASP.NET Data Binding İşlemleri

## İki tiptir:

1. Single-Value Data Binding
2. Repeated-Value Data Binding

**Not:** **Single-Value Data Binding** pek kullanışlı olmadığı için sadece Repeated-Value Data Binding'ten bahsedeceğiz.

## Repeated-Value Data Binding

**ASP.NET List Control**, **ListBox**, **DropDownList**, **CheckBoxList**, **RadioButtonList** sınıfları ile kullanılır. **Rich Web control** ise **GridView**, **DetailsView**, **FormView**, **Listview** sınıfları ile birlikte kullanılır.

### Örnek :

```
1  protected void Page_Load(object sender, EventArgs e)
2  {
3      ArrayList fruit = new ArrayList();
4      fruit.Add("Kivi");
5      fruit.Add("Elma");
6      fruit.Add("Armut");
7      fruit.Add("Tongel");
8
9      ListBox item = new ListBox();
10     item.DataSource = fruit;
11     this.DataBind();//Binding'i aktive eder. Bu metod çağrılmazsa item'ın DataSource'na fruit atanamaz.
12 }
```

## ListControl Collection

Sadece belirli türleri tutmak için kullanılan bir collection'dur. System.Collections.Generic namespace'i içerisinde bulunur.

### Örnek :

```
1  ListBox item = new ListBox();
2  List<string> liste = new List<string>();
3  liste.Add("Muz");
4  liste.Add("Elma");
5  liste.Add("Armut");
6  item.DataSource = liste;
7  this.DataBind();
```

**Not:** **DataBinding** olmasaydı List control'lerine item eklerken foreach loop kullanacaktık, yani tek tek item'ları eklemekle uğraşacaktık.

Şu alanlarda kullanılabilir : Bir form doldururken next dediği zaman **CheckBoxList**'in **DataSource**'una item'ları ekler ve **CheckBoxList**'in **Visible** property'sini true yaparız.(next butonuna tıklanmadan önce **Visible** property'yi false yaptık)

## DataBinding with a Dictionary Collection

İki temel dictionary collection türü vardır :

1. **HashTable** : **System.Collections namespace** içerisinde bulunur. Herhangi bir türden nesnelere kaydetmeyi sağlar.
2. **Dictionary** : **System.Collections.Generic namespace** içerisinde bulunur. **ArrayList** veya **List** gibi **Dictionary** nesnesi yaratılır ; fakat **Dictionary** nesnesi yaratılırken her item unique bir index ile eklenir.

### Örnek :

```
1  Dictionary<int, string> fruit = new Dictionary<int, string>();
2  fruitim.Add(1, "Elma");
3  fruitim.Add(1, "Armut");
4  fruitim.Add(1, "Tongel");
5
6  ListBox item = new ListBox();
7  item.DataSource = fruitim;
8  item.DataTextField="value";
9  this.DataBind();
```

### Örnek:

```
1  Dictionary<string, string> sozluk = new Dictionary<string, string>();
2  sozluk.Add("come", "gelmek");
```

```

3 | sozluk.Add("go", "gitmek");
4 | sozluk.Add("take", "almak");
5 | sozluk.Add("borrow", "odunc almak");
6 | ListBox item1 = new ListBox();
7 | item1.DataSource = sozluk;
8 | item1.DataTextField = "key";
9 | ListBox item2 = new ListBox();
10 | item2.DataSource = sozluk;
11 | item2.DataTextField = "value";
12 | this.DataBind();

```

**item1** nesnesinde sözlük'ün key'leri yani come,go,take,borrow ; **item2** nesnesinde sözlük'ün value'leri yani gelmek,gitmek,almak,odunc almak gözükecektir.

**Not:** **ListItem** ile **DataValueField**'in birlikte kullanılmasını araştır.

**DataTable** Class'ı Kullanarak Tablo Yaratmak:

1. **DataTable** nesnesi yaratılır.
2. **DataColumn** nesneleri yaratılır. **DataTable** nesnesine eklenir.
3. **DataTable** nesnesinin **DataRow()** metoduyla row'lar yaratılır. **DataTable** nesnesine eklenir.
4. **DataSet** nesnesi yaratılır. Bu nesneye yaratılan **DataTable** nesnesi eklenir.

**Örnek:**

```

1 | protected void Page_Load(object sender, EventArgs e)
2 | {
3 |     DataColumn c1mID = new DataColumn("ID");
4 |     DataColumn c1mName = new DataColumn("Name");
5 |     DataTable tblYeni = new DataTable();
6 |     tblYeni.Columns.Add(c1mID);
7 |     tblYeni.Columns.Add(c1mName);
8 |     DataRow rwIlk = tblYeni.NewRow();
9 |     rwIlk["ID"] = "1723413212";
10 |    rwIlk["Name"] = "Muhammed";
11 |    tblYeni.Rows.Add(rwIlk);
12 |    DataSet yeniDataSet = new DataSet();
13 |    yeniDataSet.Tables.Add(tblYeni);
14 |    ListBox1.DataSource = tblYeni;
15 |    ListBox1.DataTextField = "Name";
16 |    this.DataBind();
17 | }

```

**Not:** **DataTextField** **ListBox**'ın **DataSource**'una bir tablo atandığı zaman veya bir **Dictionary** nesnesi olduğu zaman kullanılır ; çünkü hem tablo da hem de **Dictionary** nesnesinde farklı sütunlar vardır. **DataTextField** 'e bir tablonun sütun ismini atarsak o tablonun ilgili sütunundaki değerler **ListBox**'ta gösterilir. **Dictionary** nesnesinde ise key ve value olmak üzere iki değer vardır. **DataTextField**'e "key" string'i atarsak **ListBox**'ta key'ler ; "value" string'i atarsak **Dictionary** nesnesinin value'leri atanır.

**Not:** Eğer **ListBox**'ın **DataSource** property'sine bir **DataSet** nesnesi atanırsa **ListBox**'ın **DataMember** property'sine de **DataSet** nesnesindeki tablo isimlerinden biri atanmalıdır. **ListBox**'ın **DataTextField**'ine ise **DataMember**'a atanan tablo'nun bir sütun'unun ismi atanır.

**Örnek :**

```

1 | DataColumn c1mID = new DataColumn("ID");
2 | DataColumn c1mName = new DataColumn("Name");
3 | DataTable tblYeni = new DataTable("Tablomuz");
4 | tblYeni.Columns.Add(c1mID);
5 | tblYeni.Columns.Add(c1mName);
6 | DataRow rwIlk = tblYeni.NewRow();
7 | rwIlk["ID"] = "1723413212";
8 | rwIlk["Name"] = "Muhammed";
9 | tblYeni.Rows.Add(rwIlk);
10 | DataSet yeniDataSet = new DataSet();
11 | yeniDataSet.Tables.Add(tblYeni);
12 | ListBox1.DataSource = yeniDataSet;
13 | ListBox1.DataMember = "Tablomuz";
14 | ListBox1.DataTextField = "Name";
15 | this.DataBind();

```