

ASP.NET Web Uygulamalarında Kullanıcı Girişlerini Kontrol Etmek

Cengiz HAN

cengiz@cengizhan.com

www.cengizhan.com

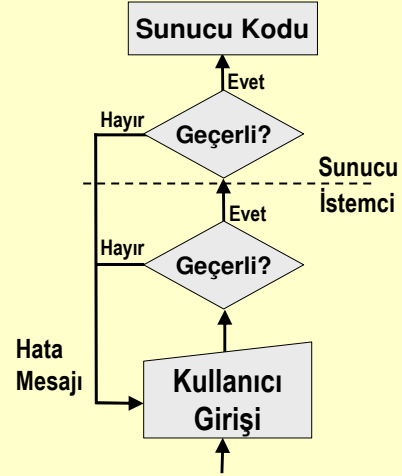


Kullanıcı Girişlerini Kontrol Etmek

- Konu 1: Kullanıcı Girişlerini Tanımak
- Konu 2: Geçerlilik Kontrollerini Kullanmak
- Konu 3: Sayfa Geçerliliği
- Gözden Geçirme
- Labaratuar

Konu 1: Kullanıcı Girişlerini Tanımak

- Nedir?
- İstemci Tarafı ve Sunucu Tarafı



3

ASP.NET Geçerlilik Kontrolleri

1. CompareValidator	Karşılaştırma kontrolü
2. CustomValidator	Özel geçerlilik kontrolü
3. RangeValidator	Aralık kontrolü
4. RegularExpressionValidator	Regular Expression kalıpları ile kontrol
5. RequiredFieldValidator	Girdi gerekliliği kontrolü
6. ValidationSummary	Hata mesajlarının özeti

4

Konu 2: Geçerlilik Kontrollerini Kullanmak

1. Geçerlilik kontrolü Web Form'una eklenir.
2. Geçerlilik kontrolü yapılacak olan giriş kontrolü belirlenir. (ControlToValidate özelliği)
3. Geçerlilik kontrol özellikleri belirlenir.
4. Hata mesajı belirlenir.
 - ErrorMessage özelliği
 - Text özelliği

Geçerlilik Kontrolleri Sözdimi

```
<asp:kontrol_ismi id="kontrol_id" runat="server"
ControlToValidate="dogrulanacak_kontrol_id"
ErrorMessage="Uyarı olarak verilecek mesaj metni!!!"
Display="Static|Dynamic|None"
Text="Kontrolün yanında gözükecek mesaj!">
</asp:kontrol_ismi>
```

5

Display Özelliği

•Tasarım Zamanı

•Çalışma Zamanı

•Static

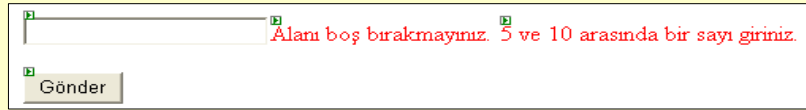
•Dynamic

- Hangi özellik seçilmeli?

6

Birden Fazla Geçerlilik Kontrolünün Bir Sunucu Kontrolü ile İlişkilendirilmesi

- Bazı durumlarda tek geçerlilik kontrolü yetersiz kalır.
- Bir giriş kontrolü ile birden fazla geçerlilik kontrolü ilişkilendirilebilir.



7

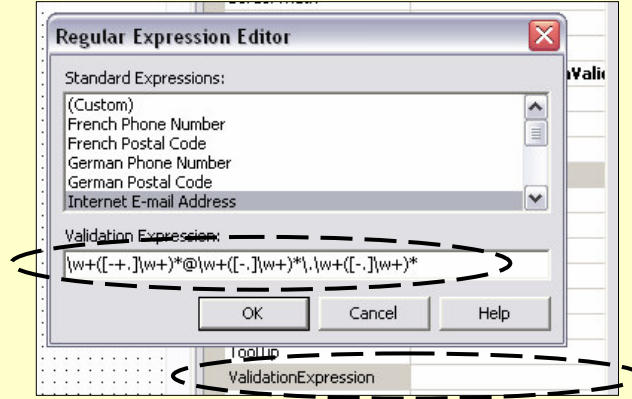
Geçerlilik Kontrolleri

- **RequiredFieldValidator**
 - InitialValue özelliği
- **CompareValidator**
 - ControlToCompare veya ValueToCompare seçimi
 - Type : Veri türü.
 - Operator : Karşılaştırma operatorü.
- **RangeValidator**
 - MaximumValue
 - MinimumValue
 - Type

8

RegularExpressionValidator Kullanımı

- Kullanıcı girdisinin özel kalıplara uyumu için kullanılır.
- Visual Studio .NET hazır olarak çok kullanılan kalıpları sunmaktadır.



9

RegularExpression Kalıpları

a	Küçük a harfi kullanılmalıdır. Önünde \ karakteri olmayan harfler yazılması gereken harflerdir.
1	1 sayısı kullanılmalıdır. Önünde \ karakteri olmayan sayılar yazılması gereken sayılardır.
?	0 veya 1 öge.
*	0 veya daha fazla öge.
+	1 veya daha fazla öge.(En az 1)
[0-n]	0 dan n' e kadar olan sayılar.
{n}	Uzunluk n karakter olmalıdır.
 	Birden fazla kalıp arasında ayırıcı görevi görür.(veya)
\	Sonraki karakterin komut karakteri olduğunu belirler.
\w	Bir karakter kullanılmalıdır.
\d	Bir sayı kullanılmalıdır.
\.	Bir nokta karakteri kullanılmalıdır.

10

E-mail Kontrolü için RegEx Örneği

`\w+([-+.] \w+)*@ \w+([-.] \w+)*\ . \w+([-.] \w+)*`

<code>\w+</code>	En az bir karakterden oluşan bir yazı.
<code>([-+.] \w+)*</code>	Bir yada daha fazla tire veya nokta, ve en az bir karakterlik yazı.
<code>@</code>	Bir @ işareti.
<code>\w+</code>	En az bir karakterden oluşan bir yazı.
<code>([-.] \w+)*</code>	Bir tire veya nokta, ve en az bir karakterlik yazı.
<code>\.</code>	Bir nokta.
<code>\w+</code>	En az bir karakterden oluşan bir yazı.
<code>([-.] \w+)*</code>	Bir tire veya nokta, ve en az bir karakterlik yazı.

- Visual Studio .NET çok kullanılan hazır kalıplar içerir.
- www.regexlib.com adresli sitede çok sayıda hazır regex kalıpları mevcut.

11

CustomValidator Kullanımı

- Kendi kontrol mekanizmanızı kodlamanızı sağlar.
- Sunucu Tarafli
- İstemci Tarafli
- ClientValidationFunction
- <CustomValidator>.ServerValidate olayı

```
Private Sub CustomValidator1_ServerValidate(ByVal source As System.Object, _  
ByVal args As System.Web.UI.WebControls.ServerValidateEventArgs) _  
Handles CustomValidator1.ServerValidate  
    Dim sayi As Integer = args.Value  
    If sayi Mod 2 = 0 Then  
        args.IsValid = True  
    Else  
        args.IsValid = False  
    End If  
End Sub
```

Kontrol
Geçerli

Kontrol
Geçersiz

```
<script language="javascript">  
function SayiCiftMi(source, arguments) {  
    var sayi = arguments.Value;  
    if (sayi % 2 == 0) {  
        arguments.IsValid = true;  
    } else {  
        arguments.IsValid = false;}}  
</script>  
</HEAD>  
<body>  
    <form id="Form1" method="post" runat="server">  
        <asp:TextBox id="TextBox1" runat="server"></asp:TextBox>  
        <asp:CustomValidator id="CustomValidator1" runat="server"  
        ErrorMessage="Sayı Çift Değil!" ControlToValidate="TextBox1"  
        ClientValidationFunction="SayiCiftMi"></asp:CustomValidator>  
    </form>  
</body>
```

12

Konu 3: Sayfa Geçerliliği

- Page.IsValid özelliği : Sayfada bulunan tüm geçerlilik kontrollerinin geçerliliğini kontrol eder. Ona göre değer alır.
- ValidationSummary Kontrolü : Sayfadaki hata veren tüm geçerlilik kontrollerinin hata mesajlarını toplu olarak gösterir. Mesaj kutusu veya metin olarak.

```
Private Sub btnKaydet_Click(ByVal sender As System.Object, _  
ByVal e As System.EventArgs) Handles btnKaydet.Click
```

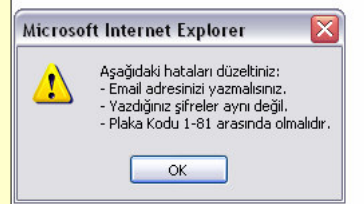
```
    If Page.IsValid Then
```

```
        lblMesaj.Text = "Sayfa geçerli durumda!"
```

```
        'işlenmesi gereken kodlar buraya yazılır.
```

```
    End If
```

```
End Sub
```



Aşağıdaki hataları düzeltiniz:

- Email adresinizi yazmalısınız.
- Yazdığınız şifreler aynı değil.
- Plaka Kodu 1-81 arasında olmalıdır.

13

Gözden Geçirme

Aşağıdaki durumlarda kullanılması uygun olan geçerlilik kontrolleri nelerdir?

1. Kullanıcının yaşı.
2. Telefon numarası.
3. Kullanıcının şifresi (iki kez girilmeli).
4. Tarih bilgisinin geçerli bir tarih olması.
5. Bir e-mail adresinin doğruluğu.

14