

Regex

Nedir?

Regex bir string ifadenin (metin) belirli kurallara uyumluluunu kontrol etmeye ve düzenlemeye yarar. Açılm **Regular Expressions**, Türkçesi ise **Düzenli ifadeler** olarak geçer.

Regex ile bir alana sadece harf mi girilmi, rakam m girilmi ya da eposta adresi mi girilmi kontrol edebilirsiniz.

Sentaks

```
<Regex Input=" " Pattern=" " Replace="False" MatchVariable="Match" />
```

Özellikler

Tip	Açıklama
Input	Bu alanda denetlemek istenilen metin bilgisini girilir. Direk metin olarak yazabilir yada "eval:/*/" ekinde form verisindeki herhangi bir alan alabilirsiniz. Not: "eval" ksm girilmesi gerekiyor
Pattern	Bu ksm eleme yapacağımız deseni belirlediğimiz alandır. Direk metin olarak yazabilir yada "eval:/*/" ekinde form verisindeki herhangi bir alan alabilirsiniz. Buras .NET Regex kütüphanesindeki standartlara göre girilmesi gerekmektedir. Aadaki adresler yeni bir desen tanımlamanız için faydalı olacaktır. Not: "eval" ksm girilmesi gerekiyor
Replace	Eer Input içindeki veriyi güncellemek istiyorsanız kullanabilirsiniz, true olduğunda bu fonksiyon metin içerisinde eleen alanlar, kendi ITasknn içine yazılmı deer dönen bir fonksiyonun deerini ile deitirir. Metin içerisindeki istenmeyen karakterleri deitirmek için kullanılabilir. <,>/ gibi xml'i bozan karakterleri deitirmek gibi.
MatchVariable	Eleen metin alanların deerlerini Regex task içerisinde yazılan fonksiyonlarda deiken olarak kullanmak için verilir. Varsayılan "Match" dir, istenirse deitirilebilir.

Örnekler

Girilen eposta tanımı standarda uygun olup olmadığını denetler ve doğrulama kuralı ile formun ilerletilmesini engelleyebilirsiniz

Örnek

```
<Project Name="">
  <Set Node="*/EpostaTest/EpostaKontrol">
    <Value Culture="" Target="Bool">true</Value>
  </Set>
  <Regex Input="eval:."
Pattern="^[^&lt;&gt;\s\@]+(\@[^&lt;&gt;\s\@]+(\.[^&lt;&gt;\s\@]+)+)$"
Replace="False" MatchVariable="Match">
    <Set Node="*/EpostaTest/EpostaKontrol">
      <Value Culture="" Target="Bool">false</Value>
    </Set>
  </Regex>
  <Evaluate Target="None"
Culture=""><![CDATA[*/EpostaTest/EpostaKontrol]]></Evaluate>
</Project>
```

Örnek 2

Herhangi bir metin içerisindeki "\${metin}" ekindeki karakterleri seçer. Dinamik bir içerik oluturma için kullanılabilir.

Örnek

```
<Script Name="">
  <Delete Node="Regex/Matches/Data/Template" />
  <Regex Input="eval:*/Regex/TargetString"
Pattern="eval:*/Regex/RegularExp" Replace="False" MatchVariable="Match">
    <Insert
Into="*/Regex/Matches/Data">:*/Regex/Matches/Template</Insert>
    <Set Node="*/Regex/Matches/Data/Template[last()]/MatchedString">
      <Value Culture="" Target="None">$(Match)</Value>
    </Set>
  </Regex>
</Script>
```

Örnek 3:

Replace özelliini true yaparak Regex Itasknda veri almas salanabilir. stenilirse bo bir karakterle veya istenilen karakterler ile deitirilebilir.

Örnek

```
<Script Name="">
  <Set Node="Regex/ReplacedString">
    <Regex Input="eval:*/Regex/TargetString"
Pattern="eval:*/Regex/RegularExp" Replace="True" MatchVariable="Match">
      <Character Type="Space" />
    </Regex>
  </Set>
</Script>
```

\$(Sayimi) alan deerin say veya text olduunun bilgisini ikili deerler olarak **Yanl(False)** veya **Doru (True)** olarak tutar. Bir deerin saym olduunu belirten **Regex** kodu **^[0-9]+\$** dr ve **./FisNo** alanında bulunan alan kontrol ederek say olup olmadn belirler.

Örnek

```
<SetVariable Name="$(Sayimi)">
  <Value Culture="" Target="Bool">>false</Value>
</SetVariable>
<Regex Input="eval:./FisNo" Pattern="^[0-9]+$" Replace="False"
MatchVariable="Match">
  <SetVariable Name="$(Sayimi)">
    <Value Culture="" Target="Bool">>true</Value>
  </SetVariable>
</Regex>
```

Ayrca Baknz

Value

SetVariable