

Rails

Model validations

1) Uygulamanın oluşturulması

Aşağıdaki komutlarla boş bir uygulama oluşturup, uygulama klasörünün içine giriyoruz. Diğer işlemleri bu klasör içinden gerçekleştireceğiz. «**crduygulama**» ismini kendinize göre değiştirebilirsiniz.

```
rails new crduygulama
```

```
cd crduygulama
```

2) Validation işlemleri için bir scaffold üretelim

- Aşağıdaki komutlarla /kitaps isminde restful tabanlı bir uygulama üretelim, bu uygulama aynı zamanda veri tabanında Kitap isimli bir model oluşturacak tablo isim,yazar, sayfa alanlarından oluşacak.

rails g scaffold kitap isim yazar sayfa:integer

rake db:migrate

rails s

Validations (Doğrulamalar)

- Alanın zorunlu olması
- Sadece rakam girilebilmesi (belirli aralıkta)
- Textlere a karakterden daha az, b karakterden daha fazla girilememesi
- Sadece email girilebilmesi
- Belirli aralıkta tarih girilebilmesi
- Telefon numarası formatı girilebilmesi
- Vb...

Alanın zorunlu olması

- Doğrulamalar için ***app/models/Kitap.rb*** dosyası üzerinde edit işlemleri gerçekleştireceğiz.
- Şu an dosya içeriği boş, herhangi bir validation girilmemiş, şu haliyle Kitap modeline web ara yüzünden yada konsoldan boş kayıt girebiliyoruz.

```
class Kitap < ApplicationRecord  
end
```

Alanın zorunlu olması

- Webten yada konsoldan boş kayıt girebiliyoruz.
- Aşağıdaki işlemin başarılı olduğuna ve boş kayıt girebildiğimize dikkat edelim.

```
irb(main):001:0> Kitap.create
  (0.1ms) begin transaction
  Kitap Create (8.0ms) INSERT INTO "kitaps" ("created_at", "updated_at") VALUES
  (?, ?) [["created_at", "2018-11-19 07:39:19.363705"], ["updated_at", "2018-11-
  19 07:39:19.363705"]]
  (8.9ms) commit transaction
=> #<Kitap id: 4, isim: nil, yazar: nil, sayfa: nil, created_at: "2018-11-19 07:
  39:19", updated_at: "2018-11-19 07:39:19">
irb(main):002:0> █
```

Kitaps

Isim Yazar Sayfa

[Show](#) [Edit](#) [Destroy](#)

[New Kitap](#)

Alanın zorunlu olması

validates_presence_of :isim

#üstteki satırı modele ekledik

reload!

#üstteki kod console ortamında verilecek (model değişikliklerinin geçerli olması için), ardından boş kayıt ekleme işlemi başarısız olacaktır.

```
class Kitap < ApplicationRecord
  validates_presence_of :isim
end
```

Alana sadece sayısal değer girilebilmesi

- Sayfa alanına text girebiliriz oysa rakam girmemiz lazım.

validates_numericality_of :sayfa

#değişikliklerden sonra console ortamında **reload!** yapmayı unutmayın

```
validates_presence_of :isim
```

```
validates_numericality_of :sayfa
```


Sayısal değerlere aralık verilmesi

```
validates :sayfa, :inclusion => {:in => 10..400, :message => "aralik dıřında"}
```

Bu örnekte sayfa sayısının 10-400 arasında olmasını istiyoruz, aynı zamanda hata mesajı atıyoruz.

reload!

```
k=Kitap.create(isim:"python",sayfa:2)
```

```
k.errors.messages
```

Hata mesajını web arayüzünden sayfa sayısı 2 olan bir kayıt ekleyerekte görüntüleyebilirsiniz.

input uzunluğunu belirleme

validates :isim, :length => { :in => 3..20, :message => "yazar ismi en az 3, en fazla 20 karakter olabilir" }

Web ara yüzünden de hata aldığımıza dikkat edelim!

New Kitap

2 errors prohibited this kitap from being saved:

- Isim yazar ismi en az 3, en fazla 20 karakter olabilir
- Sayfa aralik dışında

Isim

Yazar

Sayfa

[Back](#)

Validations syntax

- Aşağıdaki iki farklı syntax te girilebilir.

validates_presence_of :isim

validates :isim, :presence=>true

Diğer validations örnekleri

- Alanın tekil olması (sosyal ağlarda kullanılan *username* gibi)
- Sadece belirli değerler girilebilmesi (cinsiyet için **m** yada **f** girilmesi)
- Bazı alanlar için yasaklı kelimeler olması (örneğin kullanıcı adı olarak admin, root girilememesi)
- Sadece email adresi girilebilmesi
- **Regex** (istenilen formatta bilgi kabul etmek)

Validations örnekleri

```
validates :username, exclusion: { in: %w(admin superuser) }
```

```
validates :email, format: { with: /^A([^\s]+)@((?![a-z0-9]+\.)+[a-z]{2,})\z/i, on: :create }
```

```
validates :age, inclusion: { in: 0..9 }
```

```
validates :first_name, length: { maximum: 30 }
```

```
validates :age, numericality: true
```

```
validates :username, presence: true
```

```
validates :username, uniqueness: true
```

```
validates :gender, inclusion: %w(male female)
```

```
validates :password, length: 6..20
```