Arkadaşlar merhaba,

Bu videomuzda ise model alanları ile işlemler yapacağız. Konu ile ilgili özet bilgileri videoya ek olarak ekledim.

Öncelikle yeni bir model oluşturalım. Modelimizin ismi LearningFields olsun.

Bunun için model klasörünün altında bir adet learning\_fields.py dosyası oluşturuyorum. Daha önce söylediğim gibi modellerimiz Model sınıfndan türetilmek zorundadır.

from django.db import models  
  
class LearningFields(models.Model):

Modelimizi \_\_init\_\_.py dosyasından tanıtalım.

from .learning\_fields import LearningFields

Şimdi serverımızı çalıştıralım. Gördüğünüz gibi hata ile karşılaşıldı. Çünkü LearningFields modelimizi tamamlamadık. Bir class oluşturdunuz fakat sonra kullanmak istiyorsunuz diyelim. Bu gibi durumlarda hata vermemesi için pass ile geçebilirsiniz. Serverımızı tekrar çalıştırıp deneyelim. Evet serverımız sorunsuz çalıştı.

Model alanları Field sınıfından türetilmektedir. Oluşturduğumuzun alanlar Field sınıfının özelliklerini taşıyacaktır. Field sınıfından alabildiği parametrelere bakalım.

def \_\_init\_\_(self**,** verbose\_name=None**,** name=None**,** primary\_key=False**,** max\_length=None**,** unique=False**,** blank=False**,** null=False**,** db\_index=False**,** rel=None**,** default=NOT\_PROVIDED**,** editable=True**,** serialize=True**,** unique\_for\_date=None**,** unique\_for\_month=None**,** unique\_for\_year=None**,** choices=None**,** help\_text=''**,** db\_column=None**,** db\_tablespace=None**,** auto\_created=False**,** validators=()**,** error\_messages=None):

Tüm alanlar bu parametreleri alabilmektedir. Fakat bazıları için sınırlandırma getirilmiş olabilir veya bazı alanlar bunlar haricinde farklı parametreler alabilir. Örneğin IntegerField sınıfında max\_length kullanıldığında uyarı vermektedir. IntegerField için max\_length ekleyip nasıl bir uyarı verdiğine bakalım.

tamsayi = models.IntegerField(max\_length=**10**)

max\_length integer field ile kullanıldığında yok sayılmaktadır. İPUCU : Alandan max\_length silidiniz.

**WARNINGS:**

**learning.LearningFields.tamsayi: (fields.W122) 'max\_length' is ignored when used with IntegerField.**

**HINT: Remove 'max\_length' from field**

verbose\_name alanın admin sayfasıi form gibi çıktılarda görüntülecek adıdır. Eğer girilmezse alanadi kullanılır.

null argümanı eğer True ise bu alan veritabanında boş değer alabilir. Aksi taktirde alana bir değer girilmesini zorunlu tutar.

blank argümanının veritabanı alanı ile herhangi bir ilgisi yoktur. Form validasyonu için kullanılır. Eğer True ise forma boş değer girilmesi izin verir. Aksi taktirde değer girmeye zorlar.

default argümanına girilen değer eğer alanın değeri boş ise veritabanına eklenecek olan varsayılan değerdir.

unique argümanı eğer True ise alanın benzersiz olduğunu belirtir. O alan aynı değeri sadece bir kez alabilir.

primary\_key argümanı eğer True ise alanın birincil anahtar olduğunu belirtir. Bir modelde sadece bir tane alan primary\_key olarak işaretlenebilir. Eğer birden fazla alan işaretlenirse hata verir. Eğer hiçbir alan primary\_key olarak seçilmez ise Django otomatik olarak ID alanını primary\_key olarak kendi oluşturur. Birincil anahtara sahip alanlar aynı zamanda benzersiz alanlardır.

editable argümanı eğer False seçilirse ilgili alan admin paneli ve formlarda görüntülenmez. Varsayılan değeri True’dur.

help\_text argümanı alanın formlarda görüntülecek olan ilave yardım metnidir.

db\_index argümanı index oluşturulması istenilen alanlar için True seçilir.

db\_colum argümanı ilgili alanın veritabanında kullanılacak ismini ayarlar. Eğer girilmez ise alan adi kullanılır. Hatırlarsanız Product modelimizi oluştururken Django nun verdiği yerine kendi tablo ismimizi kullanmak için db\_name kullanmıştık. Buda aynı onun gibi veritabanındaki sütun ismini değiştirir.

unique\_for\_date argümanı girilen alan için o tarih ile birlikte benzersiz kayıt oluşturulmasını sağlar.

Örnek vermek gerekirse username alanına unique\_for\_date = ‘date\_joined’ verdiğimizde Django aynı tarih ve isme sahip başka birinin kayıt olmasına izin vermez. unique\_for\_month ve unique\_for\_year kullanımları da benzerdir. Bu argümanların kullanılabilmesi için modelde bir tane de tarih alanı olması gerekmektedir.

validators argümanı ilgili alan için çalıştırılacak validasyon listesini ayarlar. error\_messages argümanı ise hata oluştuğunda kullanılacak varsayılan mesajları ayarlamaya yarar. Bunları validasyon bölümünde daha ayrıntılı anlatacağım.

max\_length argümanı alanın alabileceği maksimum karakter uzunluğunu ayarlar.

choices argümanı alana daha önceden tanmlanmış seçimli öğeler atamak için kullanılır. Bu argümanı girildiğinde ilgili alan için form tarafında seçim kutusu (selectbox) görüntülenir. Şimdi model tarafında nasıl tanımladığımıza bakalım. . Bunun için model klasörünün altında bir adet learning\_choices.py dosyası oluşturuyorum. Yeni bir model oluşturalım. Modelimizin Car olsun.

from django.db import models  
  
  
class Car(models.Model):  
 pass

Modelimizi \_\_init\_\_.py dosyasında tanıtalım.

from .learning\_choices import Car

Car modelimizin alanlarını belirleyelim. Arabalarım modelleri, markaları ve üretim yılları olur. Bu alanlar şimdilik bizim için yeterli. Bunlardan markaları choices olarak ayarlaylım ve marka olarak BMW, AUDI, VOLVO ve MERCEDEES kullanılabilsin. Öncelikle seçeneklerimizi oluşturalım. Seçenekler tupple olarak oluşturulur. Seçim kutusunu düşünürseniz her bir seçinin key ve değeri oluyor. Bizde tupple ile bunları tanımlıyoruz. Markalar değerler, baş harfleri de keyimiz oluyor. Veritabanına keyleri kayıt edeceğiz.

BRAND\_CHOICES = (  
 ('A'**,** 'AUDI')**,** ('B'**,** 'BMW')**,** ('M'**,** 'MERCEDES')**,** ('V'**,** 'VOLVO')  
)

Evet marka seçeneklerini oluşturduk. Şimdi bunu bir alana kayıt etmemiz lazım. Keylerimiz text olduğu için CharField alanını kullanabiliriz. Tüm keyler bir harften oluştuğu için max\_length olarak 1 verelim ve seçim yapılmaz ise default olarak ‘A’ yani AUDI seçilmiş olsun.

brand = models.CharField(max\_length=**1,** choices=BRAND\_CHOICES**,** default='A')

Araba model için yine bir CharField oluşturalım max\_length ide 50 ile sınırlayalım.Üretim yılı içinde sayı alanı oluşturmalıyız.

model = models.CharField(max\_length=**50**)  
year = models.IntegerField()

Choices Model kısmında nasıl oluşturulacağını öğrenmiş olduk. Form oluşturma bölümünde choices için örnekler yapacağız. **get\_alan\_adi\_display()**

Alanlara verilebilecek genel argümanların çoğunu öğrenmiş olduk. Şimdide alan türlerine bakalım. Ben anlatacağım alan türlerini daha önceden oluşturdum anlatımı bunun üzerinden gerçekleştireceğim.

Alan türlerine başlamadan önce veritabanında alanların alabileceği min – max değerleri öğrenelim. Small Integer -32.768 – 32.767 arası tam sayı alabilir. Integer -2.147.483.648 – 2.147.483.647 arası tam sayı alabilir. Big Integer ise yaklaşık -9 x 1018 ie 9 x 1018 arası tam sayı alır. Positive olanlar ise eksi değer alamazlar aldıkları değerler 0 dan başlar.

BigIntegerField zaten bütün değerleri alabiliyor. Neden hepsinde bunu kullanmıyoruz diye aklınıza soru gelebilir. SmallIntegerField her bir alan için 2 byte yer kaplarken BigIntegerField 8 byte yer kaplamaktadır. Gereksiz yere BigIntegerField kullanmak veritabanınıj büyükmesine neden olur. Veritabanlarımız büyüdükçe sorgu performansı da düşer. Alanın alabileceği maksimum değer hangi aralıkta kalıyorsa o alan türü kullanılmalıdır.

Daha önceki konularda model oluşturken alanlarda primary\_key argümanı girmezsek Django’nun otomatik olarak id alanı eklediğini öğrenmiştik. Django bu alanı AutoField olarak oluşturmaktadır. AutoField’ler integer alan tipindedir. Alan adını değiştirmek istersek veya AutoField aldığı değerden fazla kaydımız olacağı için BigAutoField kullanmak istersek kendimiz oluşturabiliriz. Bu alanların değerleri otomatik olarak her kayıt eklenişinde artmaktadır. Bu alanlarda primary\_key argümanının True olarak seçilmesi zorunludur.

Sayı alanlarının oluşturulmasında fark yoktur. Sadece alabileceği değerler farklı olur.

Float Field ve Decimal Field ondalık sayılar için kullanılan alan tipleridir. Bunların farkı Float Field küçük sayıları ile Decimal Fieldin daha büyük sayıları kabul etmesidir.Float Fieldin en fazla alabileceği basamak adedi 7 dir ve veritabanında 4 bye yer kaplar.Decimal Fieldin ise alabileceği en fazla basamak 29 dur. Veritabanında 16 byte yer kaplar. Decimal Fieldın diğer bir farkıda basamak sayısının ve ondalık kısımın sırlandırılabilir olmasıdır. Decimal Field max\_digits ve decimal\_places argümanlarını alır. max\_digits maksimum izin verilen basamak sayısıdır. decimal\_places ise kullanılacak ondalık basamak sayısıdır. decimal\_places, max\_digits’den büyük olamaz.

BooleanField, True / False ve None değeri alabilir. Varsayılan olarak None seçilidir.

CharField255 karaktere kadar stringler için kullanılan alan tipidir. max\_length argümanı tanımlanması zorunludur.

TextField ,Uzun stringler için kullanılan alan tipidir. max\_length argümanı seçilirse otomatik oluşturulan form için geçerli olur.

DateField datetime.date türünde tarih alanıdır. auto\_now ve auto\_now\_add argümanlarını alır. auto\_now True ise save methodu çağırıldığında alanı güncel tarih ile değiştirir. auto\_now\_add True ise nesne ilk oluşturulduğunda alana güncel tarihi ekler. auto\_now\_add, auto\_now ve default argümanı birlikte kullanılamaz.

DateTimeField, datetime.datetime türünde tarih alanıdır. DateField ile aynı özellikleri taşır.

TimeField, datetime.time türünde tarih alanıdır. DateField ile aynı özellikleri taşır.

EmailField, Sadece email adreslerini kabul eden alan tipidir.

GenericIPAddressField, protocol ve unpack\_ipv4 argümanlarını alır. protocol argümanı both, IPv4 ve IPv6 değerlerini alabilir. unpack\_ipv4 argümanı protocol sadece both ise kullanılabilir. unpack\_ipv4 default olarak False seçilidir. True seçilirse ise IP adresini verir.

SlugField, Slug oluşturmak için kullanılan alan tipidir. harf, sayı, alt çizgi ve kısa çizgi alabilir. Genellikle url için kullanılır. max\_length argümanı alır. max\_length default 50 olarak ayarlanmıştır. db\_index argümanı default olarak True ayarlanmıştır.

URLField. URL adreslerini kabul eden alan tipidir. max\_length argümanı alır. default olarak 200 seçilidir.

UUIDField, Python UUID sınıfını kullanarak 32 karakterlik benzersiz bir key oluşturur.

BinaryField, Binary datalar için kullanılan alan tipidir.

Dosya alan türleri (FileField, FilePathField, ImageField) ve ilişkisel alan türleri (ForeignKey, ManyToManyField, OneToOneField) hakkında bilgiler ileriki konularda anlatılacaktır.