Django Projesi Notları - 1

15 Ağustos 2016 Pazartesi 12:11

- Kurulum yapılır.
- Dil ayarlamalarını yapmak için settings.py dosyasının içindeki ilgili yeri aşağıdaki gibi düzenliyoruz :
  - LANGUAGE_CODE = '**tr-TR**'
- Model oluştururken null değer vermek için :
  - **(null=True, blank=False)**
    - Null -> Database boş bırakılabilir
    - Blank -> Formdan gelen değerler boş olabilir.
- **class Place (models.Model):**
  category = models.ForeignKey(Category, blank=True, null=True)
  - Bu şekilde bir yerin bir kategorisi, bir kategorinin birden fazla yeri olabilir.
  - Teke çok ilişki
- Model oluştururken türkçe karakter sorununu halletmek için :
  - **from django.utils.encoding import smart_text**
  - class Category (models.Model):
    name = models.CharField(max_length=255)

    def __str__(self):
    return smart_text(self.name)
- Django ayarlarında belirtilen user tablosunu oluşturacapımız model içerisine çekmek için :
  - class Review(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL)
    - Böylece ayrı bir user modeli oluşturmamıza gerek kalmaz.
    - Django user modeli hazır bir şablon ve tüm ilişkiler ile CRUD işlemleri otomatik oluşturulmuş.
- Resim yükleme işlemi için model içerisine oluşturulacak kod :
  - **image = models.ImageField(upload_to="uploads")**
    - Buradaki "uploads" kısmı resimlerin yükleneceği konumu ifade eder.
    - Burada resimleri imageField olarak tutmak için virtualenv klasörünün içine pillow kurmamız gerekiyor
      - **Pip install pillow**
    - Bu sayede image dosyalarını db üzerinde saklayabiliriz.
- Admin panelinden görünen isimleri değiştirmek için model kısmında şu tanımlamaları yapmamız lazım :
  - class Category (models.Model):
    name = models.CharField(max_length=255)

    class Meta:
    verbose_name_plural = "Categories" -> Çoğul isim görünümünü düzenleme
    verbose_name = "Category" -> Tekil isim görünümünü düzenleme

- Admin panelinde görüntülemek için admin.py içinde yapılması gereken düzenlemeler :

  - from django.contrib import admin
    from blog.models import (
        Place, Category, Review, Media
)

    class MediaInline(admin.TabularInline):
        model = Media #Modelin nerden geleceğini seçiyoruz
        extra = 0 #Modelin birden fazla kaç kere gelmesini istediğimizi seçiyoruz

    class PlaceAdmin(admin.ModelAdmin): #Place kısmına özellik atamak için

        list_display = ("name", "user", "category", "has_wifi") #Önyüzde gösterilmesini istediklerimizi yazıyoruz

        list_editable = ("category", "has_wifi") #Detay kısmına gitmeden direk değiştirmek istediklerimizi yazıyoruz

        inlines = [MediaInline] #Foreign key ile bağlı olan bir modeli detail sayfasında göstermek istiyorsak
        # bu yolu izliyoruz.
        # Burada ayrı bir class belirtmemiz lazım.

    admin.site.register(Place, PlaceAdmin) #Üst kısımda yazdığımız class ı buraya çekiyoruz
    admin.site.register(Category)
    admin.site.register(Review)
    admin.site.register(Media)

- Media gösterimi için yapılması gerek değişiklikler :

  - Settings.py - En alta eklenecek:
    - MEDIA_ROOT = os.path.join(BASE_DIR, "uploads")
      MEDIA_URL = "/media/"

  - Url.py - Normal bir kısma eklenecek :
    - if settings.DEBUG:
        urlpatterns += [url(r'^media/(?P<path>.*)$', serve, {
          "document_root" : settings.MEDIA_ROOT,
        })]

  - Model.py kısmındaki kodlar :
    - class Media(models.Model):
        place = models.ForeignKey(Place)
        image = models.ImageField(upload_to="places")