Kullanacağımız modelleri oluşturduk bu modellere göre yönetici panelimizi düzenleyelim. Daha önceden hatırlarsanız yönetici paneli ile temel işlemler yapmıştık. Şimdi bu bilgilerimizi biraz daha genişleteceğiz.

Öncelikli olarak proje ismini güncelleyelim. Bunun için apps.py de verbose\_name girmemiz gerekmektedir.

*project/apps.py*

from django.apps import AppConfig  
  
  
class ProjectConfig(AppConfig):  
 name = 'project'  
 verbose\_name = 'Stok Uygulaması'

Daha sonra uygulamamızın config sınıfını init dosyasında tanıtmalıyız.

*project/\_\_init\_\_.py*

default\_app\_config = 'project.apps.ProjectConfig'

Uygulamamızda yönetim panelinden düzenleme yapmak istediğimiz sınıfları yönetim paneline ekleyelim. Bu ayarları admin.py de yapıyoruz. admin.site.register(Product) yazarak yönetim paneline ekleyebiliriz.. Eğer built-in gelen User ve Group kaldırmak istersek unregister kullanabiliriz. Öncelikle modellerimizi dosyaya import edelim. Şimdi modellerimizi register edebiliriz.

*project/admin.py*

from django.contrib import admin  
from .models import Dealer**,** Distributor**,** Product**,** Order**,** Brand**,** Category, Payment  
  
  
# Register your models here.  
admin.site.register(Dealer)  
admin.site.register(Distributor)  
admin.site.register(Product)  
admin.site.register(Order)  
admin.site.register(Brand)

admin.site.register(Category)

admin.site.register(Payment)

Modellerimizdeki alanların doğru gözükmesi için verbose\_name leri ekleyelim. Eğer sınıflarda \_\_str\_\_ metodumuz tanımlanmamışsa tanımlayalım ve örnek olması açısından veritabanına Kullanıcı, marka, distribütör, bayi ve ürün ekleyelim. (BUNU SEED İLE YAPABİLİR MİYİZ?)

class ProductAdmin(admin.ModelAdmin):  
 # fields = ('name',) # formda görüntülenecek alanlar ayarlanır  
 # readonly\_fields = ('name', ) # Bu alanlar formda read only gözükür  
 exclude = ('slug'**,** ) # Bu alanlar formda gözükmez  
 list\_display = ('name'**,** 'created'**,** 'brand') # Listeleme sayfasında sütun olarak görüntülenir  
 list\_filter = ('created'**,** 'brand') # Listeleme sayfasına filtreleme olarak gelir.  
 change\_list\_template = 'admin/project/products\_change\_list.html' # tema değiştirir.  
  
  
admin.site.site\_header = 'STOK YÖNETİM PANELİ'  
admin.site.site\_title = 'bir stok yöneticisi'  
admin.site.index\_title = 'Stok Yönetimi'

# Register your models here.  
  
admin.site.register(Product**,** ProductAdmin)

{% extends 'admin/change\_list.html' %}  
  
{% block object-tools %}  
  
 Ürün yönetim sayfası.  
 {{ block.super }}  
  
{% endblock object-tools %}

Tüm modellerimiz için bu değişiklikleri yapalım.

class ProductAdmin(admin.ModelAdmin):  
 list\_display = ('name'**,** 'brand'**,** 'created'**,** 'active')  
 list\_filter = ('created'**,** 'brand'**,**'distributor'**,**'active')  
 change\_list\_template = 'admin/project/products\_change\_list.html'  
  
  
class DealerAdmin(admin.ModelAdmin):  
 list\_display = ('name'**,** 'author'**,** 'created'**,**'active')  
 list\_filter = ('created'**,** 'author'**,**'active')  
  
  
class DistributorAdmin(admin.ModelAdmin):  
 list\_display = ('name'**,** 'author'**,** 'created'**,**'active')  
 list\_filter = ('created'**,** 'author'**,**'active')  
  
  
class BrandAdmin(admin.ModelAdmin):  
 list\_display = ('name'**,**'created')  
 list\_filter = ('created'**,**)  
  
  
class OrderAdmin(admin.ModelAdmin):  
 list\_display = ('dealer'**,**'order\_transfered'**,**'order\_cancelled'**,**'created'**,**'updated')  
 list\_filter = ('dealer'**,**'order\_transfered'**,**'order\_cancelled'**,**'created')  
  
  
class CategoryAdmin(admin.ModelAdmin):  
 list\_display = ('name'**,**)

class PaymentAdmin(admin.ModelAdmin):  
 list\_display = ('distributor'**,**'dealer'**,**'amount'**,**'payment\_accepted'**,**'created')  
 list\_filter = ('distributor'**,**'dealer'**,**'amount'**,**'payment\_accepted'**,**'created')

admin.site.site\_header = 'STOK YÖNETİM PANELİ'  
admin.site.site\_title = 'bir stok yöneticisi'  
admin.site.index\_title = 'Stok Yönetimi'

# Register your models here.  
  
admin.site.register(Product**,** ProductAdmin)  
admin.site.register(Dealer**,** DealerAdmin)  
admin.site.register(Distributor**,** DistributorAdmin)  
admin.site.register(Order**,** OrderAdmin)  
admin.site.register(Brand**,** BrandAdmin)  
admin.site.register(Category**,** CategoryAdmin)  
admin.site.register(Payment**,** PaymentAdmin)

Ürün ekleme sayfasında alanları yan yana alabiliriz.

class ProductAdmin(admin.ModelAdmin):  
 fields = (('name'**,** 'active')**,** 'content'**,** ('brand'**,** 'distributor')**,** ('price'**,** 'stock\_count'))  
 list\_display = ('name'**,** 'brand'**,** 'created'**,** 'active')  
 list\_filter = ('created'**,** 'brand'**,**'distributor'**,**'active')  
 change\_list\_template = 'admin/project/products\_change\_list.html'

Bazı alanları birlikte gruplayabiliriz. Bunun için fieldsets ayarı kullanılır. class’a collapse girerek istediğimiz grupları gizleyebiliriz.

class ProductAdmin(admin.ModelAdmin):  
 fieldsets = (  
 ('Zorunlu'**,** {  
 'fields': ('name'**,** 'content'**,** 'active')  
 })**,** ('Opsiyonel'**,** {  
 'classes': ('collapse'**,**)**,** 'fields': ('brand'**,** 'distributor'**,** 'price'**,** 'stock\_count')**,** })**,** )  
 list\_display = ('name'**,** 'brand'**,** 'created'**,** 'active')  
 list\_filter = ('created'**,** 'brand'**,**'distributor'**,**'active')  
 change\_list\_template = 'admin/project/products\_change\_list.html'

Biz standart form ile devam edeceğiz. Bilgi olması açısından aktardım.

class ProductAdmin(admin.ModelAdmin):  
 list\_display = ('name'**,** 'brand'**,** 'created'**,** 'active')  
 list\_filter = ('created'**,** 'brand'**,**'distributor'**,**'active')