

Admin Sayfası Yapımı

13 Ağustos 2016 Cumartesi 15:31

PyCharm Ayarları

- ★ Model.py kısmında düzenleme yapıyoruz. Örnek kod aşağıdaki gibidir. Burası bizim db oluşturmak için kullanacağımız modeli ifade eder.

- `from django.db import models`

```
class Post(models.Model):
    title = models.CharField(max_length=100)
    body = models.TextField()
    is_published = models.BooleanField()

    if __name__ == '__main__':
        def __str__(self):
            return self.title
        # Buraya yazılan girdi ( self.title ) admin panelimizden girdiğimizde Post kısmı altında
        # görülecek alanı belirler
        # Birden fazla alan girilebilir, veya string bir değer de yazılabilir
```

- ★ Sonra setting.py altına oluşturduğumuz app adını girmemiz gerekir. Örnek kod aşağıdadır.

- ```
INSTALLED_APPS = [
 'django.contrib.admin',
 'django.contrib.auth',
 'django.contrib.contenttypes',
 'django.contrib.sessions',
 'django.contrib.messages',
 'django.contrib.staticfiles',

 'blog',
]
```

- ★ Admin.py kısmı altında düzenlemeler yapıyoruz. Bunu yapmamızın amacı, eklediğimiz class yapılarının admin panelinde gözükmesidir. Her eklediğimiz class için buraya ekleme yapmamız gerekli.

- ```
from django.contrib import admin
from blog.models import Post
```

```
admin.site.register(Post)
```

KONSOL KODLARI

- ★ Bu kodların hepsi activate kodu çalıştırdıktan sonra manage.py dosyasının olduğu klasörde yapılır.

- ★ **python manage.py makemigrations**

`makemigrations` ile veritabanı senkronizasyonu için migration dosyaları oluşturulur.

- ★ **python manage.py migrate**

`migrate` komutu ile migration'lar veritabanında uygulanır.

★ **python manage.py createsuperuser**

Bir süper kullanıcı oluşturulur.

★ **python manage.py runserver**

`runserver` ile development sunucusu başlatılır.