

# DJANGO WEB ARAYÜZÜ

Merhaba arkadaşlar, uzun süre sonra tekrar Django web arayüzü ( framework ) hakkında yeni ve güncel bilgiler paylaşmak için tekrar beraberiz. Zero2Hero şeklinde ilerleyip, herkesin anlayabileceği şekilde konuşma diliyle yazacağım. Bu sebeple noktalama işaretlerinde hatalar yapılmış olabilir.

Django nedir önce kısa bir bilgi verelim. Django; Python kodlarıyla hazırlanmış bir web arayüzüdür. Yani Php ile hazırlanan Wordpress, OpenCart, NukePHP gibi Django'da Python kodlarıyla geliştirilmiş bir araçtır diyebiliriz. Tam olarak Türkçe karşılığı terimi bulamıyoruz ancak kısa sürede web siteleri oluşturmamızı sağlar. Örnek olarak bir kaç web sitesi verebiliriz.

<http://disqus.com/>  
<http://pinterest.com/>  
<https://www.youtube.com/>  
<https://www.dropbox.com/>  
<http://instagram.com/>  
<http://support.mozilla.org/en-US/home>  
<http://dpaste.com/>  
<http://www.spormarket.com.tr>  
<http://www.grafson.com>  
<http://www.izmirteknikservis.tk>

Daha fazlası: <https://www.djangosites.org/>

Bir çok siteyi zaten tanıyorsunuz.

Bunlar size Django ile neler yapılabileceği hakkında fikirler verebilir. Python tüm işletim sistemlerinde rahatlıkla kullanıldığı için Django'yu da tüm işletim sistemlerinde kullanabilirsiniz. Symbian, ios, android, windows, raspberry pi gibi küçük bilgisayarlar üzerinde linux ve windows kurulumu yapılarakta kullanılabilir.

Her zaman söylediğim gibi **tüm yazılım dilleri aynıdır sadece syntax (yazım şekli) değişir**. Buradan yola çıkarak Django'yu gözünüzde zor diye büyütmeğe herşeyi yapabilirim diye büyütmeğe daha iyidir.

Peki Django neden bu kadar çok firma tarafından kullanılırken Türkiye'de pek fazla destek görmüyor?

Aslında görüyor ama klasik yazılım dilleri gibi yükle-kullan olarak kullanılamıyor. Klasik Asp, Php, Asp.NET gibi hosting alıp dosyalarınızı yükleyip hosting firmanız ile görüşerek hemen kullanmaya başlayamıyorsunuz. Django kullanmaya başladığınızda server (backend diyebiliriz) tarafını da sizin kontrol etmeniz gerekir. Bir çok firma artık size vps desteği

vererek ssh ile bağlanıp kendi serverınızı kullanarak Django ile çalışmanıza yardımcı oluyor. Hatta hazır Django kurulumu yaptırılabilir. Zamanında DjangoTürkiye.com'da bunu yapmıştık ama o zaman kimse Django'yu bilmiyordu, bu sebeple beklediğim gibi bir sıçrama olmamıştı.

Konuyu anlatmaya başlarken en azından bir kaç yazılım diliyle çalıştığınızı kabul ederek devam edeceğim. Hiçbir yazılım dili ile bir çalışma yapmadınız yada henüz karar vermediyseniz biraz zorlanacaksınız ama yine de fikriniz oluşacak.

Şu an Python'ı biliyor ve Django harici Python web arayüzlerini denediniz ise Django için hazırsınız diye düşünerek kurulum ve kullanıma başlayabiliriz.

Django projesi <https://www.djangoproject.com/> adresinde tanıtılmakta ve anlatılmaktadır. Sürekli gereksinimlere karşı da güncellenmektedir. İleri de sizde gruba dahil olabilirsiniz.

**Bu arada projeye bağış yapmayı unutmayın.** <https://www.djangoproject.com/fundraising/>

İlk başlayanlar veya geçiş yapanlar için gördüğüm en zor adım kurulum aşaması. Bu sebeple en uzun duracağım konu burası olacak.

Şu an son sürüm olarak **1.9 ( 1.9.1 )** hazırlandı. Önceki sürüm **1.8 ( 1.8.8 )** LTS yani uzun süre destek verilecek sürümdür. **Nisan 2018'e** kadar da destek verilecek.

Bu konu hakkında da sorular geliyor.

## Neden güncelleme yapılıyor? Neden önceden yapmadılar?

Kısa bir örnek vereyim hemen, HTML5 ile gelen yeni özellikler için bile eklemeler yapıldı. JSONField

Diğer birkaç konuya örnek gerekirse; sık kullanılan bir çok komutların tek fonksiyonda toplanması, güvenlik açıkları yamaları yada veritabanlarında yapılan değişikliklere eklenti sağlanması

Daha fazla merak edenler için:<https://docs.djangoproject.com/en/1.9/releases/>

Release Series	Release Date	End of mainstream support <sup>1</sup>	End of extended support <sup>2</sup>
1.10	August 2016	April 2017	December 2017
1.11 LTS <sup>3</sup>	April 2017	December 2017	Until at least April 2020
2.0	December 2017	August 2018	April 2019
2.1	August 2018	April 2019	December 2019
2.2 LTS	April 2019	December 2019	Until at least April 2022
3.0	December 2019	August 2020	April 2021

[1] Güvenlik düzeltmeleri ve veri kaybı hata, çökme hataları, yeni tanıtılan özellikleri önemli fonksiyonellik hataları, Django eski sürümlerinden gerilemeleri

[2] Güvenlik düzeltmeleri ve veri kaybı hata

[3] Python 2.7 ile desteklenen son sürüm

\*Daha iyi bir çeviri gerekir.

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

# KURULUM

Kurulum için farklı işletim sistemlerinde bir çok yol mevcut. Ben Ubuntu üzerinde çalışıyorum ama diğer işletim sistemlerine de değineceğim. Zaten kurulumdan sonra yazım aşamasında farklılık yok.

1. olarak indir-kur yönetimine değinelim.

- <https://www.djangoproject.com/download/1.9.1/tarball/>

Yukarıdaki linkten Django-1.9.1.tar.gz (7.1MB) adında sıkıştırılmış bir dosya indirilecek.

\* Windows kullananlar için 7zip veya Winrar bu dosyayı açacaktır.

Sıkıştırılmış dosyası açtıktan sonra aşağıdaki gibi bir görüntü oluşacak.

## **Linux:**

```
muslu@muslu-MS-7641:~/İndirilenler/Django-1.9.1$ ls
AUTHORS CONTRIBUTING.rst django Django.egg-info docs extras Gruntfile.js INSTALL js_tests LICENSE MANIFEST.in
package.json PKG-INFO README.rst scripts setup.cfg setup.py tests
```

## **sudo python setup.py install**

komutu ile kurulumu başlatabiliriz.

## **django-admin -version**

komutu ile kurulumun doğru tamamlandığı kontrol edilir.

Tek satırda yapmak isterseniz:

```
cd İndirilenler/
```

```
tar -xzf Django-1.9.1.tar.gz && cd Django-1.9.1/ && sudo python setup.py install
```

```
django-admin -version
```

**&& kullanarak sırayla komut ekleyebilirsiniz.**

Alınabilecek hatalar:

**[Errno 13] Permission denied: '/usr/local/lib/python2.7/dist-packages/.....'**

komutun başına **sudo** eklemeyi unuttunuz ve yetkiniz yok.

**sudo python setup.py install**

olmalı

### ***Windows:***

Python kurulmuş ve PATH alanına eklenmiş olduğu yani komut satırında **python** komutuna izin verilmiş olması gerekir.

### **python setup.py install**

kurulum tamamlandıktan sonra komut satırına **python** yazarak python idesi açılabilir.

```
import django  
django.VERSION  
exit()
```

komutları djagonun kurulumu ve versiyonu kontrol edilebilir. Bu komutlar diğer işletim sistemleri içinde geçerlidir.

2. olarak **pip** kurulumuna değinelim.

**pip** kısaca python modüllerini kurmamız için geliştirilmiş bir paket yöneticisi.

Ubuntu:

```
sudo apt-get install python-pip
```

Windows:

```
python -m pip install -U pip
```

Mac OS:

```
sudo easy_install pip
```

pip kurulumundan sonra pip ile kurulumlara başlayabiliriz.

Kullanım örneği:

```
sudo pip install Django  
sudo pip install Django=1.7.7
```

Mac OS:

```
sudo pip install django
```

Windows:

```
pip install django
```

Toplu kurulum için  
**sudo pip install -r requirements.txt**

Ben genelde pip'i kullanmayı tercih ediyorum. Hem eski versiyonu otomatik kaldırıyor hemde tek bir dosya oluşturup tümünü bir arada kurabiliyoruz.

Kurulum aslında bu kadar zaten bir çok arkadaşta farklı platformlarda kurulumları gayet başarılı anlattılar.

Şu an için gerekli olarak görmedeğim **virtualenv**, shell kullanımı gibi konulara girmeyeceğim.

Kontrolleri sağladıktan sonra **Django** artık hazır, sizi bekliyor.

Django'nun bilinen bir açığı yok yani sayfanızın hacklenme gibi bir şey söz konusu değil. Tabi yazılımcı hataları olmazsa. Zaten html dosyasına yönlendirme sizin elinizde olduğu için biri gelip index.htm\* dosyalarının tümünü değiştirtse bile sayfanıza bir şey olmayacaktır.

Eğer bir hata alıyorsanız bu güzel bir şey demek ve mutlaka birileri bunu tecrübe edinmiş, önlemini almış ve açıklamasını yapmış. Mümkün olduğunca hatalar alıp bunların açıklamalarını yapacağım.

Okunabilirliği artırmak için yazı boyutlarını büyük tutup sık boşluk kullanmaya çalışıp, çok terim ve uzun cümleler kullanmaktan kaçındım sanırım böyle daha anlaşılır olmuştur.

## Düzenleme Aracı

Düzenleme aracı olarak ( editör ) ben **JetBrains**'e ait **Pycharm** ( Professional) kullanıyorum. İlk videolarda **Gedit** kullanmıştım ama artık **Pycharm** ile anlatacağım.

JetBrains editör konusunda çok başarılı. Tüm ürünlerini denedim ve fiyatları da gerçekten çok uygun.

<https://www.jetbrains.com/pycharm/download/#section=linux>

buradan kendi işletim sisteminize göre Pycharm'ı indirebilirsiniz. Otomatik olarak linux gelecek.

Hemen bir proje oluşturarak artık çalışmaya başlayabiliriz ama bazı terimleri şimdiden anlatmak gerekiyor.

Proje: **Wikipedia**' da “bir probleme çözüm bulma ya da beliren bir fırsatı değerlendirmeye yönelik, bir ekibin, başlangıcı ve bitişi belirli bir süre ve sınırlı bir finansman dahilinde, birtakım kaynaklar kullanarak, müşteri memnuniyetini ve kaliteyi göz önünde bulundururken olası riskleri yönetmek şartıyla, tanımlanmış bir kapsama uygun amaç ve hedefler doğrultusunda özgün bir planı başlatma, yürütme, kontrol etme ve sonuca bağlama sürecidir” diye tanımlanıyor.

Sürekli duyduğumuz bu terim aslında “**bir fikrim var**” yerine kullanılıyor. Oysa ki fikir henüz başlanmamış, eyleme geçilmemiş ve akılcılıkla ilgilidir. Oysaki proje zamanı belirlenmiş, başlanmış, planlanmış ve ekip olarak tasarlanmış fikirler ve eylemler.

Burada da proje; başlangıç olarak yapmayı istediğimiz web sitesinin genel adıdır.

Örnek olarak; teknik servis takibi, sağlık ocağı sıra takibi, kombin ürün satış sitesi vs..

## Artık başlayalım:

```
mkdir django
cd django
django-admin startproject teknikservistakibi
cd teknikservistakibi/
ls -la
```

projemize ait bir klasör oluşturu ve içinde **manage.py** dosyası ve proje adı ile aynı bir klasör daha oluşturuldu.

\* Django'nun eski versiyonlarında bu klasör oluşturulmuyor ve dosyalar direk dışarıda tutuluyordu.

\* Proje oluşturmayı Pycharm'dan da yapabiliriz ama komut olarak öğrenmeniz daha iyi. Çünkü her zaman bir editörünüz olmayacak ve her zaman local de çalışamayacaksınız.

```
muslu@muslu-MS-7641:~$ tree django/
django/
├── teknikservistakibi
│   ├── manage.py
│   └── teknikservistakibi
│       ├── __init__.py
│       ├── settings.py
│       ├── urls.py
│       └── wsgi.py
```

2 directories, 5 files

\* **tree** komutu için **sudo apt-get install tree**

**manage.py:** Proje ve uygulamalar ile ilgili komutları çalıştıracığımız yönetim dosyası.

**\_\_init\_\_.py:** Genel kullanımı bu klasörde python dosyaları var demek ( Python paketlerini içeren dizinler ) ama yine bir py dosyası olduğu için içine özel komutlar yada açıklamalar eklenebilir.

**settings.py:** Adından da anlaşılacağı gibi ayarların bulunduğu dosya. Projenin tüm ayrıntıları burada. Wordpress'teki config.php gibi..

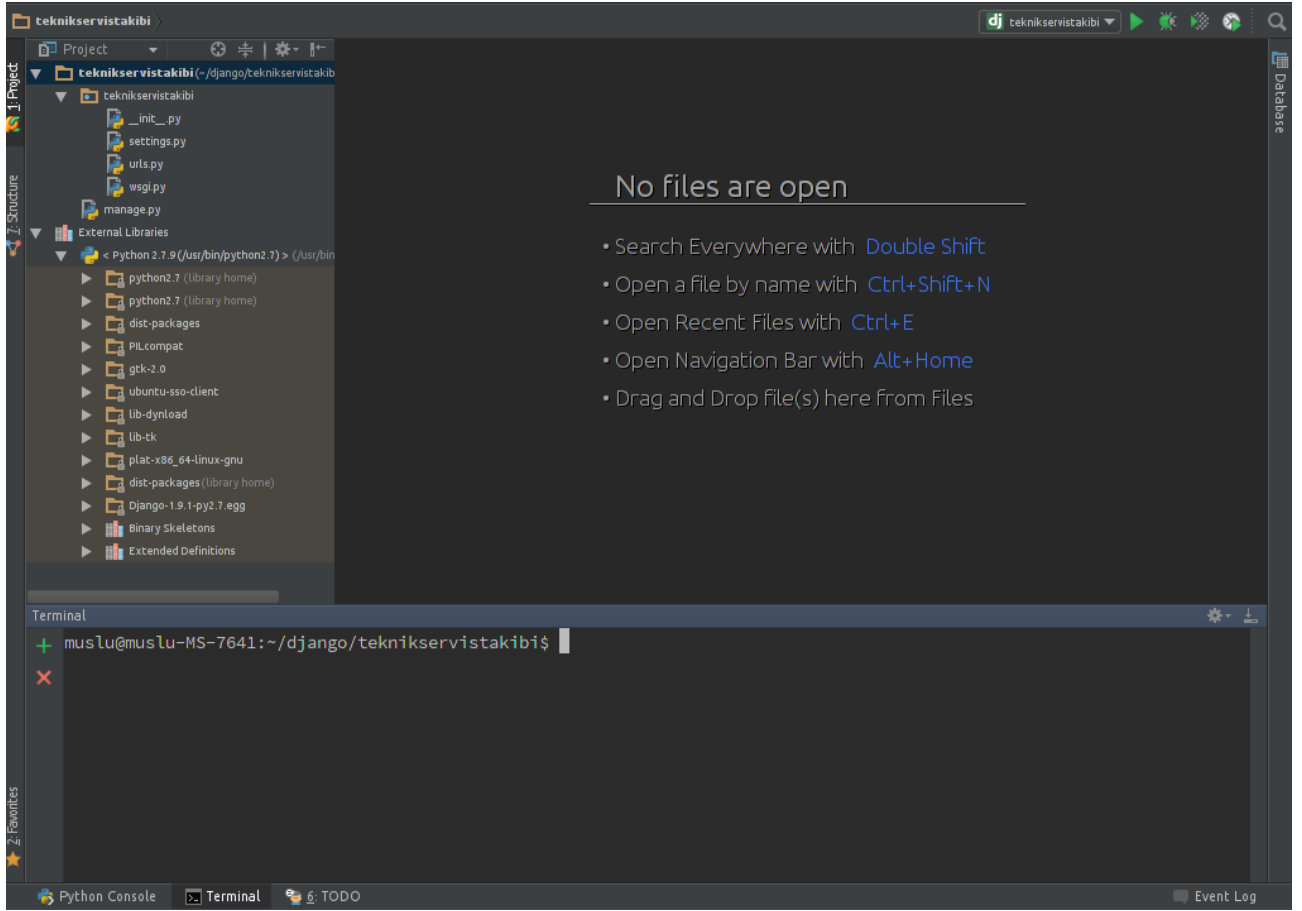
**urls.py:** Url yönlendirmelerinin yapılacağı dosya. Aynı zamanda fonksiyon da yazabiliriz.

**wsgi.py:** Http serverlar ( Örneğin **Apache** ve **Nginx**. \*libapache2-mod-wsgi ) için proje yönlendirme dosyası diyebiliriz, şu an çok detaya girmeye gerek duymuyorum.

İlk projemiz oluşturuldu. Şimdi bir kaç ayar yaparak ilk testi yapabiliriz.

Pycharm'ı başlatarak gelen ekrandan **Open** ile projenizin klasörünü (/django/teknikservisformu/) seçiyoruz.

\* Alt+F12 ile terminali açabilirsiniz.



Settings.py de ufak bir kaç ayar yaparak Türkçeleştirme yapıyoruz.

ctrl+g

Satır: 107-109

```
LANGUAGE_CODE = 'tr_TR'  
TIME_ZONE = 'Europe/Istanbul'
```

Terminalde;

**python manage.py makemigrations && python manage.py migrate**

yazarak değişiklikleri onaylatıp, hata olup olmadığını kontrol ediyoruz.

\* 1.8 den sonra syncdb artık tamamen kullanılmıyor.



## python manage.py runserver

komutu ile Django'nun basit bir http serverını çalıştırıyoruz.

```
System check identified no issues (0 silenced).  
January 12, 2016 - 13:58:43  
Django version 1.9.1, using settings 'teknikservistakibi.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.
```

Burada karşılaşılabilecek hatalara değinelim.

### Error: That port is already in use.

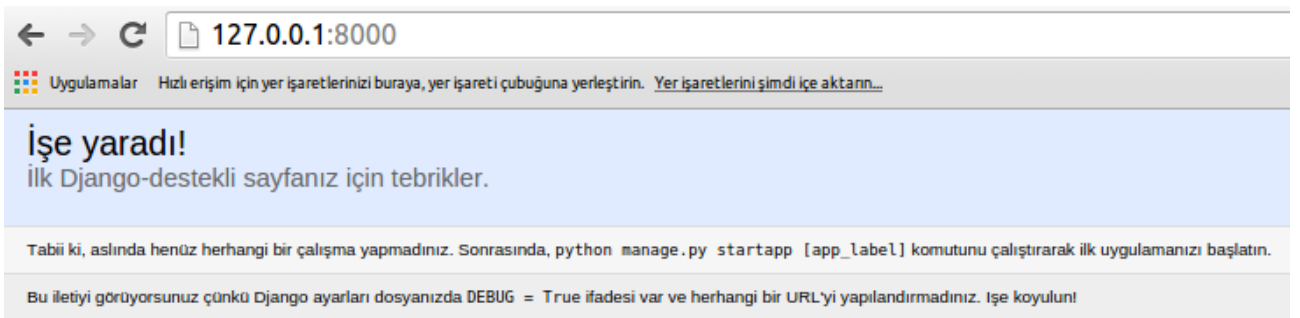
Eğer farklı bir komut satırında aynı proje ve/veya farklı bir proje çalışıyorsa bu hatayı alırsınız. Yani bu port zaten kullanılıyor.

İlla 2 proje çalıştırılması gerekiyorsa

## python manage.py runserver 127.0.0.1:8001

ile farklı bir port üzerinden çalıştırılabilir.

Diğer bir hususta; ip adresi alan başka bir cihazdan ( pc, telefon, tablet vs.. ) projenizi kontrol etmek isterseniz komut satırınızda ip adresinizi öğrenerek bu ip üzerinden yayın yapabilirsiniz. Böylelikle farklı tarayıcılarda nasıl gözüktüğüne bakabilirsiniz.



muslu@muslu-MS-7641:~/django/teknikservistakibi\$ ifconfig

```
muslu@muslu-MS-7641:~/django/tekniksर्वistakibi$ ifconfig
eth0      Link encap:Ethernet  HWaddr d4:3d:7e:51:d4:3b
          inet addr:192.168.2.168  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::d63d:7eff:fe51:d43b/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10623053  errors:0  dropped:0  overruns:0  frame:0
          TX packets:13422150  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:2270926581 (2.2 GB)  TX bytes:12170714775 (12.1 GB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Sunucu
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:44712  errors:0  dropped:0  overruns:0  frame:0
          TX packets:44712  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:0
          RX bytes:4300496 (4.3 MB)  TX bytes:4300496 (4.3 MB)

muslu@muslu-MS-7641:~/django/tekniksर्वistakibi$
```

Diğer alınabilecek hatalar;

Port u belirtilmezseniz alacağınız hata

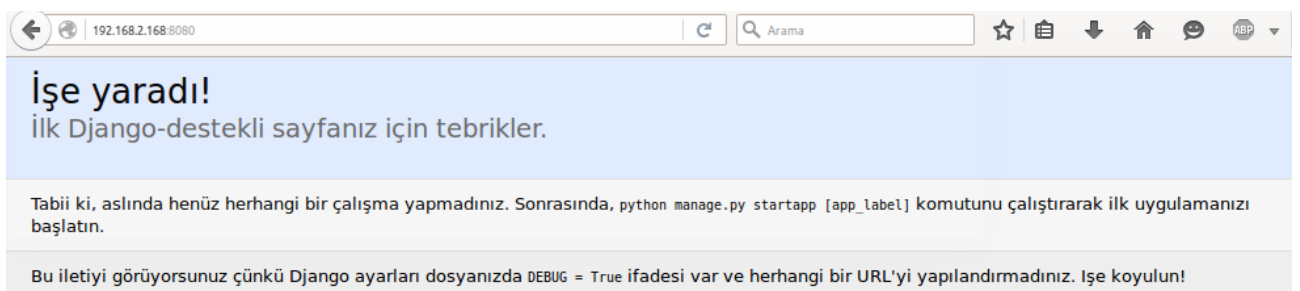
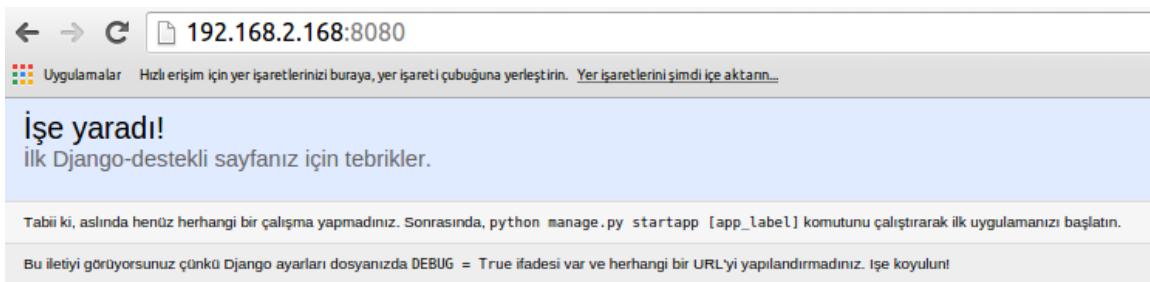
**CommandError: "192.168.2.168" is not a valid port number or address:port pair.**

Port adresi belirttiğiniz halde farklı bir komut satırında yada bir http server ( apache ) yüklü ise yani bu port kullanılıyorsa.

**Error: You don't have permission to access that port.**

Münasip bir port bulduğunuzda deneme yapabilirsiniz.

**python manage.py runserver 192.168.2.168:8080**



Gördüğünüz gibi **Django** projemiz çalışmaya başladı ve **Türkçe** olarak yayınlanıyor.

Setting.py dosyamızdan dili değiştirip test edebilirsiniz. Neredeyse bilinen tüm dillere destek veriyor.

Alınabilecek diğer hata ise:

**CommandError: You must set settings.ALLOWED\_HOSTS if DEBUG is False.**

Eğer DEBUG modundan çıkmak isterseniz, yani hataların apaçık şekilde yayınlanmasını istemiyor, özelleştirilmiş bir html dosyasında gösterilmesini istiyorsanız ALLOWED\_HOSTS listesine kabul edilen ip ve adresleri yazmanız gerekir.

Örnek:

```
ALLOWED_HOSTS = ['192.168.2.168', '127.0.0.1', '.izmirteknikservis.tk']
```

www kullanmanız gerektiğinde **.domain.uzantisi** şeklinde yazabilirsiniz. İleride daha detaylı değineceğiz.

Değişikliği yaptıktan sonra artık bir sayfa gelmeyecek ve Not found uyarısı verecektir. Çünkü url olarak herhangi bir yönlendirme yapmadık.

Settings.py ile ilgili bir kaç noktaya daha değinelim ama ihtiyaç oldukça gerekli eklemeleri yapacağız.

```
import os

# Projenin bulunduğu klasöre ulaşmak için değişken
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
# benim proje yolum. BASE_DIR = /home/muslu/django/teknikservisformu/

# Hataların ekrana yansıtılması
DEBUG = False

# Çalışılacak domain isimler listesi
ALLOWED_HOSTS = ['192.168.2.168']

# Veritabanı seçimi, ayarları
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

# Kurulu uygulamalar. Yazdığımız uygulamaların listesi. Öncelik sırası var.
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

```

    ]

# Tüm projede geçerli olacak kodlar. Sıralamaya göre öncelik middleware lerdedi.
MIDDLEWARE_CLASSES = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.auth.middleware.SessionAuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

# urls.py dosyası
ROOT_URLCONF = 'teknikservistakibi.urls'

# Html dosyaları içinde gönderilecek veriler, ayarlar vs..
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

# Http serverlar için wsgi dosya adı ve uygulaması
WSGI_APPLICATION = 'teknikservistakibi.wsgi.application'

# Yetkilerde geçerli olan şifreleme yöntemleri
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

```

```

# Uluslararasılaşma
# https://docs.djangoproject.com/en/1.9/topics/i18n/
LANGUAGE_CODE = 'tr_TR'
TIME_ZONE = 'Europe/Istanbul'
USE_I18N = True
USE_L10N = True
USE_TZ = True

```

# Statik (CSS, JavaScript, Resimler) dosyaların çağıracağı url yolu. Fiziksel yol değildir dikkat edin. <http://127.0.0.1:80/static/> çağırıldığında css, js,

```
resim vss gibi dosyalara ulaşılacak.  
# https://docs.djangoproject.com/en/1.9/howto/static-files/  
STATIC_URL = '/static/'  
  
# Gizli kod. İleride gerekecek.  
SECRET_KEY = '9f5$6e&r&x3*_a%j1ocv*p3aftgkl1y5n&)+^jehhc@&z%@@@8p'
```

okunabilirlik için **boşluklar** ekledim ama bu yazım hali **PEP** standartlarına uygun değil zaten **Pycharm** da bu konuda uyaracaktır ancak **hata** olarak değil **uyarı** olarak.

**Yazım aşamasında iken Debug modunu True yapmanız gerekir.**

Alınacak hata:

**SyntaxError: Non-ASCII character '\xc4' in file /home/muslu/django/teknikservistakibi/teknikservistakibi/settings.py on line 3, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for details**

Yani diyor ki; setting.py dosyamızın 3. satırında pep standartlarına uygun olmayan kodlama sorunu var. ( non-ascii dediği ) Yorum satırı olsa bile türkçe karakter kullanamayız.

# Projenin bulunduğu klasöre ulaşmak için değişken

Bu hata ile sık karşılaşacağız, bu sebeple her dosyanızın başına **# -\*- coding: utf-8 -\*-** (Bu dosyanın kodlama şekli utf-8 dir) eklemeniz gerekiyor. Her zaman birinci satıra eklenmesi gerekir.

Django'nun bir güzel tarafı da yönetim panelinin hazır gelmesi. Kullanıcılar, gruplar ve bunların yetkileri için auth modülü bizim için hazırlanmış.

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    .....,  
]
```

\* Bonus: `django.contrib.admin` aslında `/usr/local/lib/python2.7/dist-packages/Django-1.9.1-py2.7.egg/django/contrib/admin/` fiziksel yolundaki dosyaları eklemek ( import ) demek.

Urls.py dosyasını açtığınızda göreceğiniz gibi admin sayfasına ait url aktif geliyor.

Test etmek için tarayıcınızda <http://192.168.2.168:8080/admin/> adresini açabilirsiniz.

Tasarım bozuk geldi, çünkü static dediğimiz dosyalar yüklenmedi.

Komut satırından ( terminal ) kontrol edebiliriz.

```
[12/Jan/2016 15:47:41] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 1697
[12/Jan/2016 15:47:41] "GET /static/admin/css/base.css HTTP/1.1" 404 99
[12/Jan/2016 15:47:41] "GET /static/admin/css/login.css HTTP/1.1" 404 100
```

admin sayfasındaki statik dosyalarını kendi projemize aktararak istediğimiz gibi düzenleyebiliriz.

Bunun için;  
terminalde ctrl+c ile çalışan komutu durdurup,

**python manage.py collectstatic**

yazmamız gerekir ama hata alacağız. Çünkü **STATIC\_ROOT** tanımlamasını yapmadık.

**django.core.exceptions.ImproperlyConfigured: You're using the staticfiles app without having set the STATIC\_ROOT setting to a filesystem path.**

Yani; **STATIC\_URL** isteği ile gelen linkin **fiziksel** karşılığını yazmalıyız. **/static/** olarak gelecek soruya, cevap olarak proje klasörümüzün altındaki static klasörünün ( izin demeyi pek tercih etmiyorum ) fiziksel yolunu vermek.

Settings.py dosyamızda aşağıdaki tanımlamaları yapıyoruz.

```
STATIC_ROOT      = BASE_DIR + "/static/"
STATIC_URL       = '/static/'
```

ek olarak projemizin medya dosyaları için ayrı bir klasör oluşturarak admin statik dosyalarından ayırmamız daha iyi olacak.

```
MEDIA_ROOT      = BASE_DIR + '/media/'
MEDIA_URL       = '/media/'
```

```
URL              = Fiziksel yol
http://192.168.2.168:8080/static/      = /home/muslu/django/teknikservisformu/static/
http://192.168.2.168:8080/media/ css/stil.css = /home/muslu/django/teknikservisformu/media/css/stil.css
```

terminalden yada dosya yöneticisi ile static ve media adında klasörlerimizi oluşturalım.

\* static klasörünü oluşturmasanız bile collectstatic komutu oluşturacak.

**mkdir static**

**mkdir media**

Tekrar collectstatic ile statik dosyaları projemize kopyalabiliriz.

**python manage.py collectstatic**

Uyarı olarak /home/muslu/django/teknikservistakibi/static klasörüne kopyalanacak ne dersiniz diye soruyor.

yes

**ls static/**

diyerek yada proje klasörümüzdeki static klasörüne bakarak admin klasörünün oluşup oluşmadığını kontrol edebiliriz.

**python manage.py runserver 192.168.2.168:8080**

ile tekrar projemizin yayınını başlatalım ve tarayıcıda <http://192.168.2.168:8080/admin/> sekmemizi yenileyelim.

Alınabilecek hata:

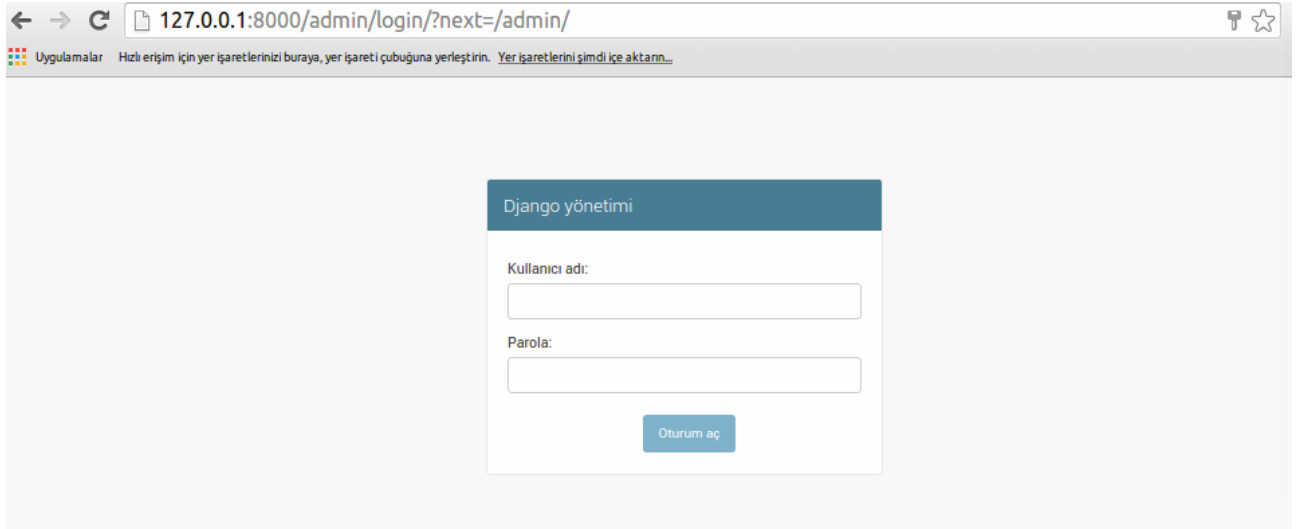
**Bad Request (400)**

Debug modu hala False!

**Statik dosyalar hala 404 veriyor.**

Debug modu hala False.

Herhangi bir sorun yaşamadınız ise aşağıdaki gibi yönetim paneli gelecek.



Yönetim paneline giriş yapabilmemiz için super yetkili bir kullanıcı oluşturmalıyız.

Daha sonra super olmasa da yetkili kişilerin girmesi için panelden kullanıcı oluşturacağız.

\*Django 1.8 öncesinde **syncdb** ile yetkili kullanıcıda oluşturabiliyorduk ama artık komut ile oluşturmamız gerekiyor.

Terminalde ctrl+c ile yayını durdurup

**python manage.py createsuperuser**

komutu ile super kullanıcı oluşturabiliriz.

Sorulara cevap verdikten sonra yetkili kullanıcı oluşturabilirsiniz. Bu komutu unutmayın ileride şifreyi unutursanız başvuracaksınız.  
Alınabilecek hatalar:

**This password is too short. It must contain at least 8 characters.**

**This password is entirely numeric.**

Şifreniz çok kısa. En az 8 karakter olması gerekiyor.



**This password is too common.**

Klasik bir şifre seçimi yapıldı.

**The password is too similar to the email address.**

Email adresi ile benzer şifre seçildi.

**Error: Your passwords didn't match.**

Yazılan bilgiler aynı değil.

Bu parola doğrulama ve oluşturma seçenekleri 1.9 ile geldi.

<https://allmychanges.com/p/python/django/>

Settings.py de ki **AUTH\_PASSWORD\_VALIDATORS** listesinden istediğiniz ( istemediğiniz ) kontrol şekillerini kaldırabilirsiniz.

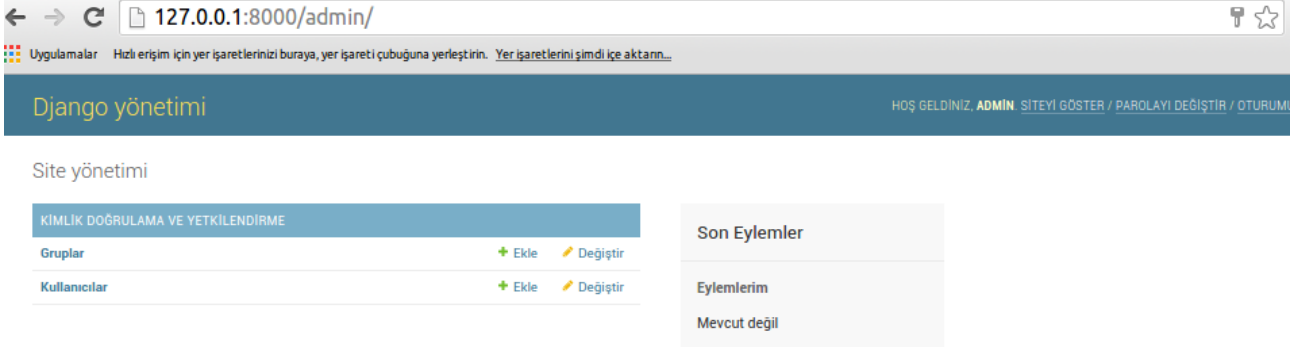
```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    # {
    #     'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    # },
    # {
    #     'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    # },
]
```

Uygun bir şifre seçtikten sonra

**Superuser created successfully.**

Bilgisini alacağız.

Yönetim sayfamızı açarak <http://127.0.0.1:8000/admin/login/?next=/admin/> giriş yapabiliriz.



Yönetim panelimiz açıldı.

13 Ocak 2016

Django'nun kendine has yönetim paneli temasını aslında beğenmiyorum, ileri de **suit** panelinin kurulum ve kullanılmasına değineceğim.

Önce admin paneli temasını nasıl değiştiririz konusuna bakalım.

Django'nun admin panel dosyalarını kendi projemizdeki templates klasörüne taşıdığımızda (statik dosyaları gibi) istediğimiz gibi düzenleme yapabiliriz.

Proje klasörümüzde **templates** adında bir klasör oluşturalım.

Bu klasörde **html**, **txt** ve **xml** dosyalarımızı saklayacağız.

index.html, robots.txt, sitemap.xml, vs..

Bu templates klasörümüzün Django tarafından geçerli olması için de settings.py dosyamızda düzenleme yapmamız gerekiyor.

Templates ayar listemizdeki DIRS değişkenini proje klasörümüzün altındaki templates klasörü olarak güncelliyoruz.

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        .....,
    },
]
```

Şimdi Django'nun kurulduğu klasör yolunu bulalım.

Terminal açarak aşağıdaki komutu yazın.

Linux, Mac OS, Windows:

```
python -c "import django; print(django.__path__)"
```

Bonus: python -c "....." ile python kodlarını çalıştırabilirsiniz.

Bonus: "... ; ....." ile tek satırda kod yazabilirsiniz.

Windows:

C:\Python27\lib\site-packages\django\contrib\admin\templates\

Dosya yöneticiniz ile bu klasörü açarak **admin** klasörünü proje klasörünüzdeki **templates** klasörünü kopyalamanız yeterli.

Linux, Mac OS:

['/usr/local/lib/python2.7/dist-packages/Django-1.9.1-py2.7.egg/django']

İsterseniz dosya yöneticiniz ile kopyalamayı yapabilirsiniz

/usr/local/lib/python2.7/dist-packages/Django-1.9.1-py2.7.egg/django/contrib/admin/templates/admin klasörünü proje klasörünüzün altında oluşturduğunuz templates klasörüne kopyalayabilir

yada terminalde;

```
cd ~/django/teknikservistakibi/templates
```

komutu ile **proje klasörümüzdeki templates** klasörüne geçiyoruz.

```
cp -a /usr/local/lib/python2.7/dist-packages/Django-1.9.1-py2.7.egg/django/contrib/admin/templates/admin/. !
```

**Dikkat: 2 adet nokta var. Birisi admin/ klasörünün altındaki tüm dosyalar demek, diğeri bulunduğumuz klasör demek.**

Hata almaktan çekiniyor yada anlaşılmadıysa alttaki komutları kullanabilirsiniz.

- cd /usr/local/lib/python2.7/dist-packages/Django-1.9.1-py2.7.egg/django/contrib/admin/templates/
- cp -a admin/ ~/django/teknikservistakibi/templates/

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

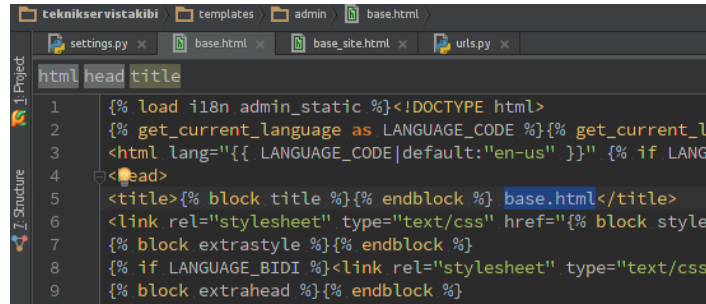
Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

- `cd ~/django/teknikservistakibi/templates/`
- `cd` ile djangonun admin templates klasörüne gittik
- `cp -a` ile admin ve alt klasörlerini proje klasörümüzdeki templates klasörümüze kopyaladık.
- `cd` ile tekrar proje klasörümüzdeki templates klasörümüze döndük.

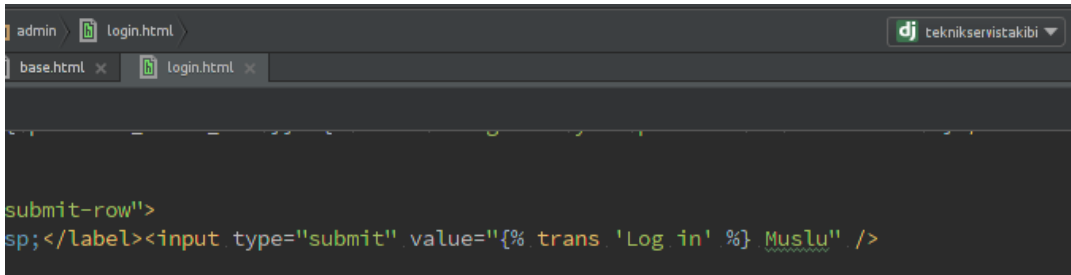
`ls`, `tree` veya dosya yöneticiniz ile kopyala yapılıp yapılmadığını kontrol edebilirsiniz.

Kopyalamayı tamamladıktan sonra deneme yapabiliriz.

Pycharm da templates admin `base.html` ve `base_site.html` dosyalarını açalım ve title etiketlerini değiştirelim.



```
1 {% load i18n admin_static %}<!DOCTYPE html>
2 {% get_current_language as LANGUAGE_CODE %}{% get_current_la
3 <html lang="{% LANGUAGE_CODE|default:'en-us' %}" {% if LANGU
4 <head>
5 <title>{% block title %}{% endblock %} base.html</title>
6 <link rel="stylesheet" type="text/css" href="{% block styles
7 {% block extrastyle %}{% endblock %}
8 {% if LANGUAGE_BIDI %}<link rel="stylesheet" type="text/css"
9 {% block extrahead %}{% endblock %}
```

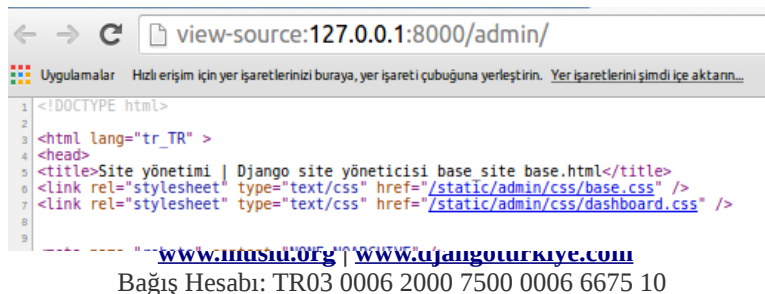


```
submit-row">
sp;</label><input type="submit" value="{% trans 'Log in' %} Muslu" />
```

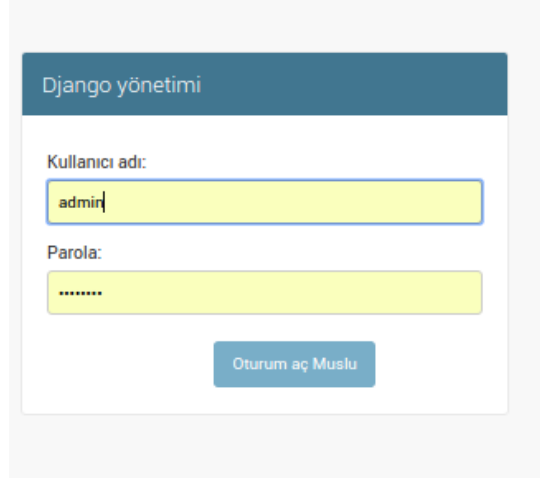
`python manage.py runserver` ile serverı başlatalım.

<http://127.0.0.1:8000/admin/>

tarayıcınızın üst kısmına baktığımızda yada kaynak kodları kontrol ettiğimizde değişikliklerin çalıştığını göreceğiz.



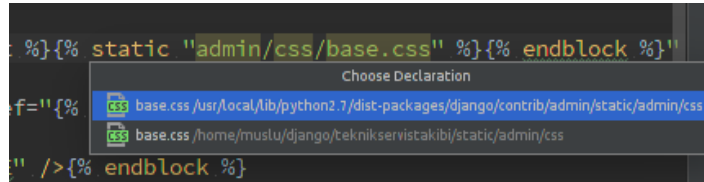
```
1 <!DOCTYPE html>
2
3 <html lang="tr_TR" >
4 <head>
5 <title>Site yönetimi | Django site yöneticisi base site base.html</title>
6 <link rel="stylesheet" type="text/css" href="/static/admin/css/base.css" />
7 <link rel="stylesheet" type="text/css" href="/static/admin/css/dashboard.css" />
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
24
```



Bundan sonra istediğiniz gibi değişiklikleri yapabilirsiniz. İleri seviyelerde `debug_tool` kullanarak yayınlanan sayfaları görerek hangi sayfaları değiştirmemiz gerektiğini de öğreneceğiz.

**Bonus:** Pycharm'da `ctrl` tuşuna basılı tutarken bir fonksiyon, script yolu, stil classı vs.. tıkladığınızda dosya yolu gösterir yada direk fonksiyonun bulunduğu dosyaya götürür.

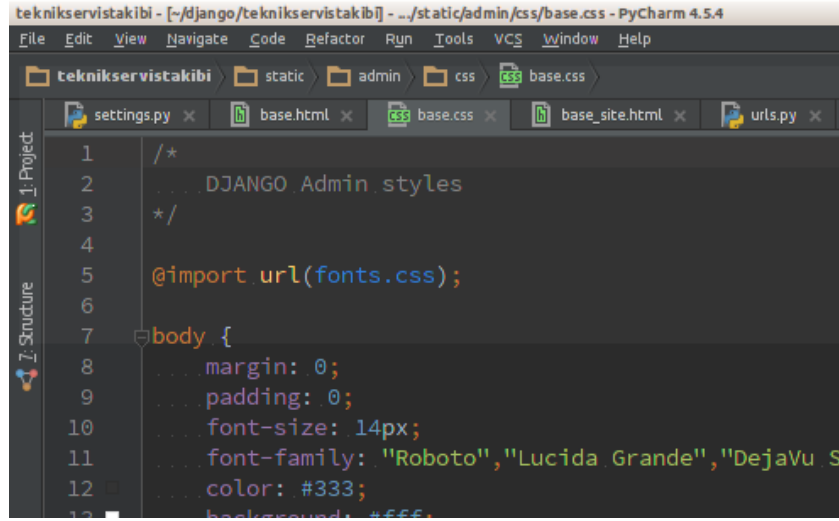
`base.css` metni üzerinde `ctrl` tuşuna basılı tutarak fare sol tuşu ile tıkladığınızda aşağıdaki gibi yolları gösterecektir ve hangisi tıklarsanız o dosyayı açacaktır.



Burada görüldüğü gibi bu stil dosyası 2 yerde mevcut.

`Collectstatic` komutu ile statik dosyalarını static klasörümüze kopyalamıştık.

Biz static klasörümüzdekileri kullanmak istiyoruz ancak bu konuya daha sonra Apache ile değineceğiz. Yönetim paneline ait html dosyaları değiştirmeniz yeterli.

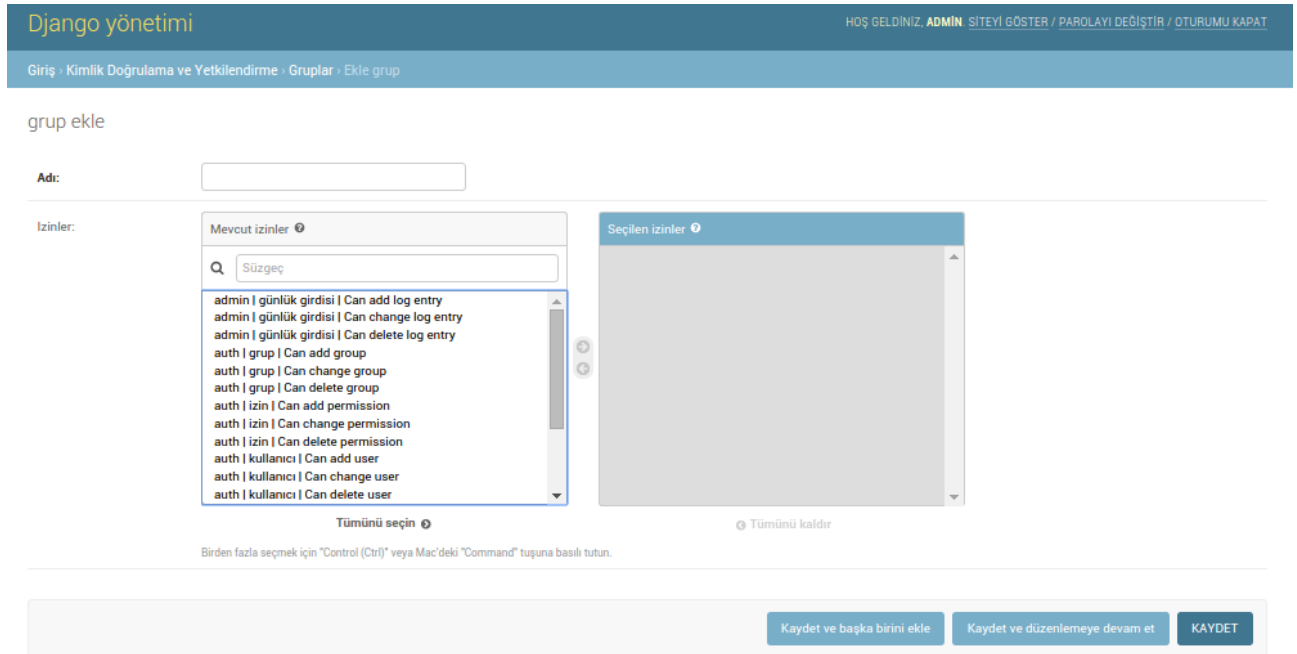


```
1  /*
2     ... DJANGO Admin styles
3  */
4
5  @import url(fonts.css);
6
7  body {
8     margin: 0;
9     padding: 0;
10    font-size: 14px;
11    font-family: "Roboto", "Lucida Grande", "DejaVu S
12    color: #333;
13    background: #fff;
```

Şimdi kısaca hazır gelen kimlik doğrulama ve yetkilendirme uygulamalarına bakıp bir uygulama yazmaya başlayalım.

Gruplar --> Ekle linki ile grup ekleme sayfasını açıyoruz. Grup adının altında ManyToMany ile çekilmiş verileri görüyoruz. Burada tüm uygulama yetkileri mevcut. Yazacağımız uygulamalarda burada gözükecek.

Örnek olarak finans, teknik servis, kargo, satış vs.. gibi gruplar oluşturulup kişilere ekleme, silme ve/veya düzenleme yetkileri toplu olarak verilebilir.



Django yönetimi

HOŞ GELDİNİZ: [ADMIN](#) [SİTEYİ GÖSTER](#) / [PAROLAYI DEĞİŞTİR](#) / [OTURUMU KAPAT](#)

Giriş > Kimlik Doğrulama ve Yetkilendirme > Gruplar > Ekle grup

grup ekle

Adı:

İzinler:

Mevcut izinler

Seçilen izinler

Tümünü seçin

Tümünü kaldır

Birden fazla seçmek için "Control (Ctrl)" veya Mac'deki "Command" tuşuna basılı tutun.

Kaydet ve başka birini ekle Kaydet ve düzenlemeye devam et KAYDET

## Kullanıcı Ekleme:

[Giriş](#) · [Kimlik Doğrulama ve Yetkilendirme](#) · [Kullanıcılar](#) · [Ekle kullanıcı](#)

kullanıcı ekle

Önce, bir kullanıcı adı ve parola girin. Ondan sonra, daha fazla kullanıcı seçeneğini düzenleyebilirsiniz.

Kullanıcı adı:

Zorunlu. 30 karakter ya da daha az olmalı. Sadece harfler, rakamlar ve @/./+/-/\_ karakterleri kullanılabilir.

Parola:

Parola onayı:

Doğrulama için önceki gibi aynı parolayı girin.

Bu ekranda kullanıcı için ad ve parola girildikten sonra detaylı bilgilerin geleceği bir sayfa gelecek.

Kullanıcı adı:

Zorunlu. 30 karakter ya da daha az olmalı. Sadece harfler, rakamlar ve @/./+/-/\_ karakterleri kullanılabilir.

Parola:

**algoritma: pbkdf2\_sha256 yinelemeler: 24000 tuz: 2oLOV0\*\*\*\*\* adresleme: RE96ve\*\*\*\*\***

Ham parolalar saklanmazlar, bu yüzden bu kullanıcının parolasını görmenin yolu yoktur, fakat **bu formu** kullanarak parolayı değiştirebilirsiniz.

Kişisel bilgiler

Adı:

Soyadı:

E-posta adresi:

İzinler

☒ Etkin

Bu kullanıcının etkin olarak işlem görüp görmediğini belirler. Hesapları silmek yerine bunun işaretini kaldırın.

☐ Görev durumu

Kullanıcının bu yönetici sitesine oturum açıp açamayacağını belirler.

☐ Süper kullanıcı durumu

Bu kullanıcıya ayrı ayrı izin atmadan tüm izinlerin verilip verilmeyeceğini belirler.

Gruplar:

Mevcut gruplar ⓘ

Kullanıcı Ekle Değiştir

Kullanıcı Sil

Seçilen gruplar ⓘ

Artık bir uygulama yazmaya başlayalım.

Terminalde proje klasörünüze geçerek

`./manage.py startapp servisformu`  
`python manage.py startapp servisformu`  
`django-admin startapp servisformu`

herhangi birini yazarak uygulamayı başlatabilirsiniz.

Alınabilecek hatalar:

**CommandError: '/home/muslu/django/teknikservistakibi/servisformu' already exists**  
Uygulama zaten oluşturulmuş, farklı bir isim seçilmeli.

**CommandError: 'django' conflicts with the name of an existing Python module and cannot be used as an app name. Please try another name.**

Python modül isimleri uygulama adı olarak kullanılamaz. Örneğin; django, math

İlk uygulamanın oluşturulmasıyla proje klasörümüzde uygulamamızın adı ile bir klasör daha oluşturuldu.

```
Terminal
+ muslu@muslu-MS-7641:~/django/teknikservistakibi$ tree
.
├── db.sqlite3
├── manage.py
├── media
├── servisformu
│   ├── admin.py
│   ├── apps.py
│   ├── __init__.py
│   ├── migrations
│   │   └── __init__.py
│   ├── models.py
│   ├── tests.py
│   └── views.py
├── static
│   └── admin
│       ├── css
│       └── base.css
```



servisformu klasöründe aşağıdaki dosya ve klasörler oluşturuldu.

admin.py: Uygulamanın admin sayfasına ait ayarların yapılacağı dosya. Models.py den gelen alanları tanımlayıp, filtreleyip, kısıtlayabiliriz.

apps.py: Uygulamanın adı ve diğer ayarlarının yapılacağı dosya

\_\_init\_\_.py: Klasik python dosyası, zaten değinmiştik.

migrations: Veritabanına ait güncellemelerin ve değişikliklerin kolay kullanım için tutulacağı klasör

models.py: Veritabanında oluşturacağımız tablo ve alanları yazacağımız dosya. Herhangi bir sql bilginiz olmasa bile kolayca yönetebileceğiz

tests.py: Uygulamanın bazı testleri deneyebileceğimiz dosya

views.py: Veritabanından gelen bilgilerin yada kendi tanımladığımız değişken yada verilerin html, txt yada xml dosyalarına yönlendirileceği dosya. Ayrıca sadece ekrana bilgi de bastırabiliriz.

Url den gelen sorguların sırayla ilerlemesini şöyle anlayabiliriz.

Tarayıcıya yazılan url önce middleware da denetlenir. Bu konuda şu an uzun durmayacağız. Daha sonra urls.py ye eklediğimiz url yönlendirmesi ile view.py ye, burada da models.py den gelen veriler tekrar templates klasöründeki dosyalara yönlendirebiliriz.

Aklınızın karışmaması için grafiksel anlatmak daha doğru olacaktır.

Basit şekilde şöyle sıralanabilir.

url.py --> views.py --> models.py --> views.py --> html

**http://127.0.0.1/formlar/ --> def formlar(request) --> class FormBilgileri(models.Model) --> def formlar(request) --> formlistesi.html**

Açıklama olarakta şöyle olabilir.

Tarayıcıdan gelen link urls.py de hangi fonksiyona tanımlandı ise buradan da models.py de tanımlanan tablodanki alanlar alınıp render edilerek html dosyasına gönderilir.

Hemen bir örnek yazarak anlaşılır hale getirelim.

Pycharm da **teknikservis** klasöründen **models.py** yi açalım ve aşağıdaki gibi kodları ekleyelim.

```
# -*- coding: utf-8 -*-
```

```
### utf-8 kodlama
```

```
from django.utils import timezone
```

```
### KayıtTarihi alanımız için otomatik bugünü seçtirme fonksiyonu
```

```
from django.db import models
```

```
### djangonun hazır modelleri. Buradan bir çok hazır alanları seçebiliriz.
```

```
class Teknisyen ( models.Model ) :
```

```
### Teknisyen adında bir tablo oluşturuyoruz.
```

```
    Aktif = models.BooleanField(default = 1)
```

```
    ### Teknisyenin aktif olup olmadığını seçmek için booleanfield kullanacağız. Default olarak seçili gelecek.
```

```
    AdSoyad = models.CharField ( u'Adı Soyadı', max_length = 250 )
```

```
    ### Teknisyenin ad ve soya dını gireceğimiz en fazla 250 karakterlik bir varchar alanı. 255 karaktere kadar yazılabilir.
```

```
    KayıtTarihi = models.DateField ( u"Kayıt Tarihi", default=timezone.now)
```

```
    ### kayıt işlemi yapıldığında otomatik bugünü seçecek ve gözükmeyecek.
```

```
    def __unicode__(self):
```

```
    ### daha önceden __str__ kullanılıyordu. Models çağrıldığında burada seçilen alan ve/veya alanlar döndürülür. Birazdan göreceğiz.
```

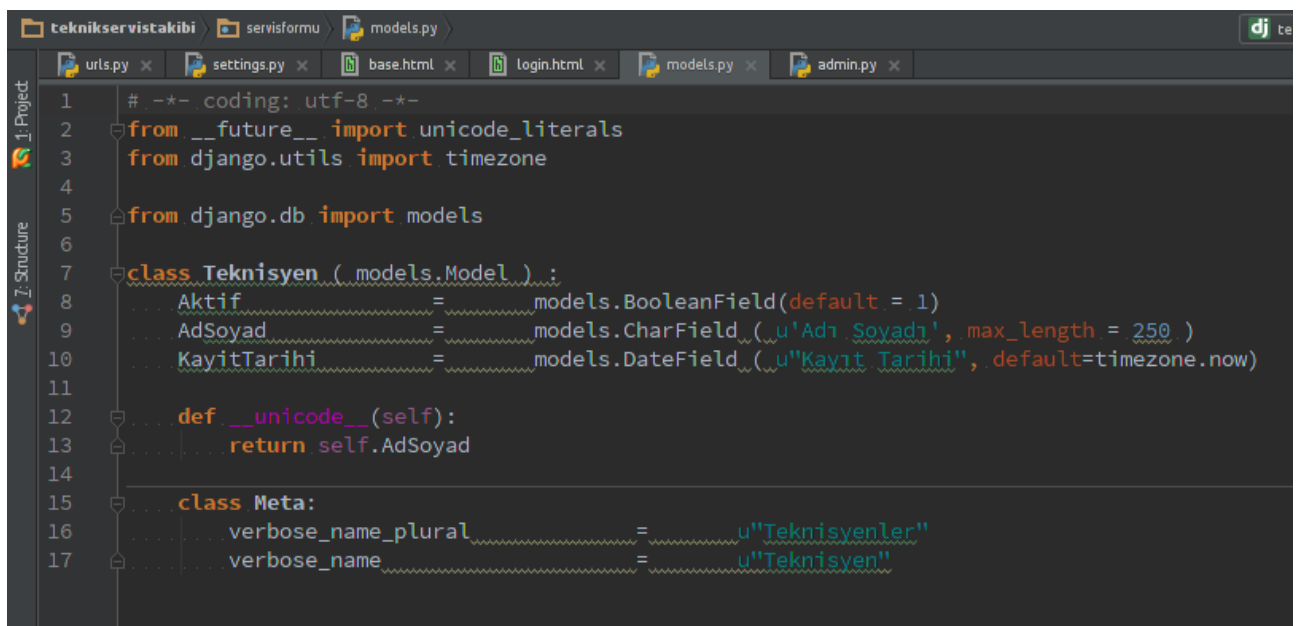
```
        return self.AdSoyad
```

```
class Meta:
```

```
### admin sayfasında bu uygulamanın nasıl isimlendirilip çağırılacağı tanımlamalar
```

```
    verbose_name_plural = u"Teknisyenler"
```

```
    verbose_name = u"Teknisyen"
```

A screenshot of a code editor window showing the Django model code for the 'Teknisyen' model. The editor has a dark theme and a sidebar on the left showing the project structure. The code is as follows:

```
1 # -*- coding: utf-8 -*-
2 from __future__ import unicode_literals
3 from django.utils import timezone
4
5 from django.db import models
6
7 class Teknisyen ( models.Model ) :
8     Aktif = models.BooleanField(default = 1)
9     AdSoyad = models.CharField ( u'Adı Soyadı', max_length = 250 )
10    KayıtTarihi = models.DateField ( u"Kayıt Tarihi", default=timezone.now)
11
12    def __unicode__(self):
13        return self.AdSoyad
14
15    class Meta:
16        verbose_name_plural = u"Teknisyenler"
17        verbose_name = u"Teknisyen"
```

\*\* kafa karıştırmamak için from \_\_future\_\_ import unicode\_literals kodlamalarına şu an değinmiyoruz.

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

admin.py dosyasını açıp aşağıdaki gibi eklemeleri yapalım.

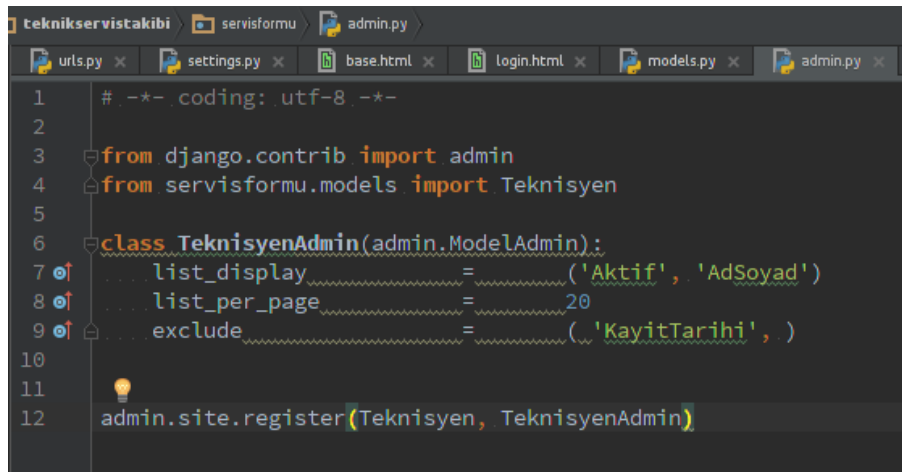
```
# -*- coding: utf-8 -*-
```

```
from django.contrib import admin
from servisformu.models import Teknisyen
```

```
### models.py dosyamızdaki Teknisyen ( class - sınıf ) tablomuzu ve alanları ekliyoruz.
```

```
class TeknisyenAdmin(admin.ModelAdmin):
    ### Admin sayfasında gösterilecek detaylar
    list_display = ('Aktif', 'AdSoyad')
    ### sırayla gösterilecek alanlar
    list_per_page = 20
    ### sayfadaki kayıt adeti, otomatik sayfalama yapacak
    exclude = ('KayitTarihi',)
    ### KayitTarihi alanını gizliyoruz
```

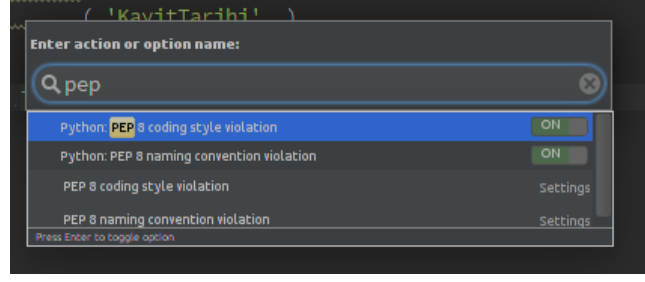
```
admin.site.register(Teknisyen, TeknisyenAdmin)
### Teknisyen ve TeknisyenAdmin sınıflarını kayıt ettiriyoruz.
```

A screenshot of a code editor window titled 'teknikservistakibi'. The editor shows the 'admin.py' file with the following code: Line 1: # -\*- coding: utf-8 -\*-; Line 2: (empty); Line 3: from django.contrib import admin; Line 4: from servisformu.models import Teknisyen; Line 5: (empty); Line 6: class TeknisyenAdmin(admin.ModelAdmin):; Line 7: list\_display = ('Aktif', 'AdSoyad'); Line 8: list\_per\_page = 20; Line 9: exclude = ('KayitTarihi',); Line 10: (empty); Line 11: (empty); Line 12: admin.site.register(Teknisyen, TeknisyenAdmin). The code is color-coded and includes a lightbulb icon on line 11.

Artık uygulamamızı projemize dahil edebiliriz.  
settings.py dosyamıza eklememizi yapalım.

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

Bonus: pep standartlarını geçici olarak pasif etmek isterseniz, ctrl+shift+a ile arama ekranını açıp pep yazabilir ve ON olan kısımları OFF yapabilirsiniz.



Terminalde;

`./manage.py makemigrations servisformu && ./manage.py migrate && ./manage.py runserver`  
yazarak yaptığımız değişiklikleri ekleyip, onaylatıp serverımızı çalıştırıyoruz.

<http://127.0.0.1:8000/admin/>



Uygulamamız yönetim sayfamızda artık hazır.

Veri ekleyip html dosyasına gönderilmeden önce yönetim sayfasında yapabileceğimiz değişikliklere bakalım.

Models.py deki

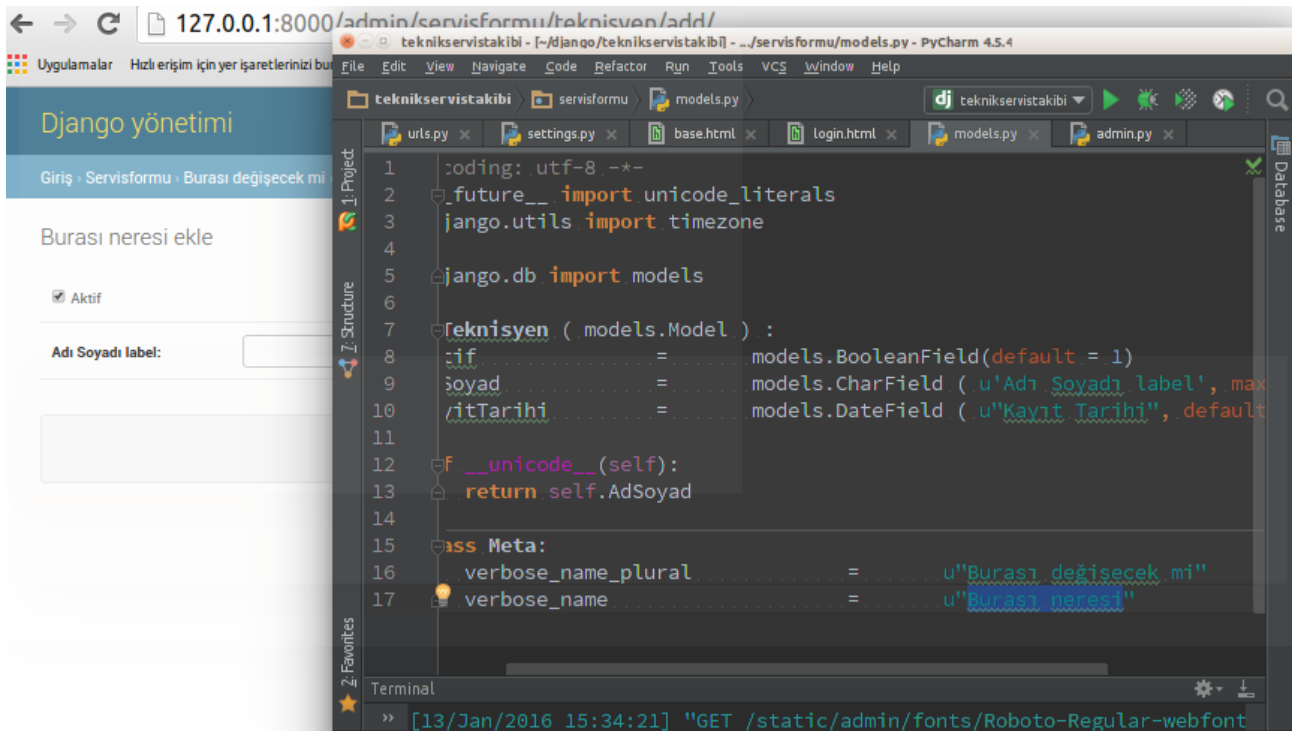
`verbose_name_plural = u"Teknisyenler"`

tanımlamamızı değiştirerek deneme yapalım.

```
11
12     def __unicode__(self):
13         return self.AdSoyad
14
15     class Meta:
16         verbose_name_plural = u"Burası değişecek mi"
17         verbose_name = u"Teknisyen"
```

verbose\_name = u"Teknisyen"

tanımlamasını değiştirerek test edebiliriz.



Değişiklikleri geri alarak kayıt girildikten sonra tüm listeyi kontrol edelim.

Teknisyen "Muslu YÜKSEKTEPE" başarılı olarak eklendi.

Bilgisini geldiği yer \_\_unicode\_\_ da eklediğimiz alanın geri dönüşü

Grid sistemindeki gelen bilgiler ise list\_display de eklediğimiz alanlar ve sıralaması ile oluşmakta.

Bu gird sistemi djangonun kendi oluşturduğu alandır ve klasik olarak tüm uygulamalarda kullanılmaktadır.

Giriş > Servisformu > Teknisyenler

✓ Teknisyen "Muslu YÜKSEKTEPE" başarılı olarak eklendi.

Değiştirmek için Teknisyen seçin

Eylem:   0 / 1 nesne seçildi

<input type="checkbox"/>	AKTİF	ADI SOYADI
<input checked="" type="checkbox"/>		Muslu YÜKSEKTEPE

1 Teknisyen

Hemen buraya kendi alanımızı ekleyelim.

```
class Meta:
    verbose_name_plural = u'Teknisyenler'
    verbose_name = u'Teknisyen'

def Yazdir ( self ) :
    return '<a href="/yazdir/%s" target="_blank">Yazdır</a>' % self.id
Yazdir.short_description = u'Yazdır'
Yazdir.allow_tags = True
```

models.py de class içine aşağıdaki gibi bir fonksiyon oluşturalım

```
def Yazdir ( self ) :
```

```
    return '<a href="/yazdir/%s" target="_blank">Yazdır</a>' % self.id
```

### Fonksiyon çağırıldığında döndürülecek metin.

```
    Yazdir.short_description = u'Yazdır'
```

### Fonksiyonun kısa açıklaması

```
    Yazdir.allow_tags = True
```

### Fonksiyonumuz html etiket içeriyor

admin.py deki list\_display oluşturduğumuz fonksiyon adını ekleyelim.

```
list_display = ('Aktif', 'AdSoyad', 'Yazdir')
```

<http://127.0.0.1:8000/admin/servisformu/teknisyen/>

Bir alan daha ekleyelim.

```
def EkAlanTest(self):
    return self.AdSoyad.replace(' ', '_____')
EkAlanTest.short_description = u'Burası alanın başlığı'
```

**list\_display = ('Aktif', 'AdSoyad', 'Yazdir', 'EkAlanTest')**

Eylem:   0 / 1 nesne seçildi

<input type="checkbox"/>	AKTİF	ADI SOYADI	YAZDIR	BURASI ALANIN BAŞLIĞI
<input checked="" type="checkbox"/>		Muslu YÜKSEKTEPE	Yazdır	Muslu_____YÜKSEKTEPE

1 Teknisyen

Alınabilecek hatalar:

**<class 'servisformu.admin.TeknisyenAdmin'>: (admin.E116) The value of 'list\_filter[2]' refers to 'Yazdir', which does not refer to a Field.**

**TeknisyenAdmin'deki list\_filter** değişken listesindeki 3. değişken geçerli bir field alanı değil.

Yani Yazdir fonksiyonu özel bir tanımlama olduğu için list\_filter da kullanılamaz.

Kayıtlara daha kolay ulaşmak ve filtreleme yapmak için Django'nun hazır bir kaç fonksiyonlarına daha bakalım...

```
list_filter = ['Aktif', 'AdSoyad', ]
### Filtre ( süzgeç ) yapabilmemiz için hazır sorgu alanı
```

```
search_fields = ['AdSoyad', ]
### Arama yapabilmemiz için otomatik bir met
in alanı oluşturur. Arayacağımız kelimeler AdSoyad'a girdiğimiz kayıtlarda sorgulayacak.
```

```
date_hierarchy = 'KayitTarihi'
### Kayıtları yıl, ay ve gün olarak otomatik filtreleme yaptırmak için gird üstünde listeleme
yapar
```

Değiştirmek için Teknisyen seçin

Q search\_fields Ara

2016 Ocak 13 date\_hierarchy

Eylem: ----- Git 0 / 1 nesne seçildi

AKTİF	ADI SOYADI	YAZDIR	BURASI ALANIN BAŞLIĞI
<input checked="" type="checkbox"/>	Muslu YÜKSEKTEPE	Yazdır	Muslu____YÜKSEKTEPE

1 Teknisyen

SÜZ

Aktif süzgecine göre

Tümü  
Evet  
Hayır

Adı Soyadı süzgecine göre

Tümü  
Muslu YÜKSEKTEPE

Şimdi de kayıt girerken yapabileceğimiz otomatik yetkilendirmelere bakalım

```
class TeknisyenAdmin(admin.ModelAdmin):
    list_display = ('Aktif', 'AdSoyad', 'Yazdir', 'EkAlanTest')
    list_per_page = 20
    exclude = ('KayitTarihi',)

    list_filter = ('Aktif', 'AdSoyad',)
    search_fields = ('AdSoyad',)
    date_hierarchy = 'KayitTarihi'

    def has_add_permission(self, request):
        return False

    def has_
admin.
    has_change_permission(self, request, obj=None)
    has_delete_permission(self, request, obj=None)
    has_module_permission(self, request)
    __hash__(self) predefined
Press Ctrl+Period to choose the selected (or first) suggestion and insert a doc afterwards

forming system checks...

tem check identified no issues (0 silenced).
uary 13, 2016 - 16:11:57
```

Giriş | Servisformu | Teknisyenler

Değiştirmek için Teknisyen seçin

Q | Ara

2016 Ocak 13

Eylem: ----- Git 0 / 1 nesne seçildi

AKTİF	ADI SOYADI	YAZDIR	BURASI ALANIN BAŞLIĞI
<input checked="" type="checkbox"/>	Muslu YÜKSEKTEPE	Yazdır	Muslu____YÜKSEKTEPE

1 Teknisyen

TEKNİSYEN EKLE +

SÜZ

Aktif süzgecine göre

Tümü  
Evet  
Hayır

Adı Soyadı süzgecine göre

Tümü  
Muslu YÜKSEKTEPE



## Kayıt Ekleme yetki fonksiyonu

```
def has_add_permission(self, request):  
    return False
```

AKTIF	ADI SOYADI	YAZDIR	BURASI ALANIN BAŞLIĞI
<input checked="" type="checkbox"/>	Muslu YÜKSEKTEPE	Yazdır	Muslu YÜKSEKTEPE

eklendiğinde süzgeç üstündeki butonun artık gelmediğini göreceksiniz.

Yani artık yeni kayıt ekleme yetkimiz yok. Bu tüm kullanıcılar için geçerlidir. Bazen tek bir kayıt olması ve bu kaydın sadece güncellenmesi gerektiğinde bu yöntemi kullanabilirsiniz. Şu an aklıma gelen bir örnek hakkımızda yazısı olabilir. Müşteriniz artık hakkımızda yazısı ekleyemez sadece düzenleyebilir ve silebilir.

Teknisyen değiştir

☒ Aktif

Adı Soyadı: Muslu YÜKSEKTEPE

Sil Kaydet ve düzenlemeye devam et KAYDET

```
def has_delete_permission(self, request, obj=None):  
    return False
```

Fonksiyonunu ekleyerek artık bu uygulama için silme yetkisini de iptal etmiş olduk.

Teknisyen değiştir

☒ Aktif

Adı Soyadı: Muslu YÜKSEKTEPE

Kaydet ve düzenlemeye devam et KAYDET

Tüm kullanıcıları değilde bazı kullanıcıları engellemek istersek, grup, üyelik, super kullanıcı yada sadece kaydı giren kişinin bu yetkilerin dışında yada içinde kalması için fonksiyona bir kaç müdahale etmemiz gerekiyor.

Aşağıdaki kodları herhangi bir fonksiyonu ekleyerek istediğimiz gibi yetkilendirme yapabiliriz ama ben has\_add\_permission için anlatacağım.

Fonksiyonlara request otomatik gönderildiği için şu an da hangi kullanıcı var öğrenebiliyoruz.

```
if not request.user.is_superuser:
### kullanıcı super inek değilse

    try:
        KayitSayisi = self.model.objects.count()
### hiç kayıt olmadığında hata verecektir.
    except:
        KayitSayisi = 0
### geçerli uygulamadaki obje sayısı, yani toplam kayıt sayısı

## KalanLimit = Uyeler.objects.get(user = request.user).KayitLimiti
### fikir olması açısından ekledim. İleri de kullanıcılara kayıt limiti verebiliriz. Grafson.com daki gibi

    KalanLimit = 5
### super olmayan kullanıcılar kayıt sayısı en fazla 5 olabilir.

    if KayitSayisi >= KalanLimit:
        return False
    else:
        return True
***return True olduğu sürece fonksiyon çalışacaktır.
```

Limit sayısını 2 yaparak deneme yapabilirsiniz ama ilk koşulu unutmayın. Yani ya super kullanıcı olmayan bir kullanıcı ile giriş yapın yada **if not request.user.is\_superuser:** satırını pasif edin.

✓ Teknisyen "Cem Emir YÜKSEKTEPE" başarılı olarak değiştirildi.

Değiştirmek için Teknisyen seçin

Q  Ara

2016 Ocak 13

Eylem:  Git 0 / 2 nesne seçildi

<input type="checkbox"/>	AKTİF	ADI SOYADI	YAZDIR	BURASI ALANIN BAŞLIĞI
<input checked="" type="checkbox"/>	✓	Muslu YÜKSEKTEPE	Yazdır	Muslu_____YÜKSEKTEPE
<input checked="" type="checkbox"/>	✓	Cem Emir YÜKSEKTEPE	Yazdır	Cem_____Emir_____YÜKSEKTEPE

2 Teknisyenler

SÜZ

Aktif süzgecine göre

Tümü  
Evet  
Hayır

Adı Soyadı süzgecine göre

Tümü  
Cem Emir YÜKSEKTEPE  
Muslu YÜKSEKTEPE

2 kayıt girdikten sonra yeni kayıt butonu pasif oldu. Eğer kaydı biri silinirse tekrar aktif olacak.

✓ 1 adet Teknisyen başarılı olarak silindi.

Değiştirmek için Teknisyen seçin

Q  Ara

2016 Ocak 13

Eylem:  Git 0 / 1 nesne seçildi

<input type="checkbox"/>	AKTİF	ADI SOYADI	YAZDIR	BURASI ALANIN BAŞLIĞI
<input checked="" type="checkbox"/>	✓	Cem Emir YÜKSEKTEPE	Yazdır	Cem_____Emir_____YÜKSEKTEPE

1 Teknisyen

TEKNİSYEN EKLE +

SÜZ

Aktif süzgecine göre

Tümü  
Evet  
Hayır

Adı Soyadı süzgecine göre

Tümü  
Cem Emir YÜKSEKTEPE

Yönetim sayfasında yetkilendirme ve kısıtlama işlemleri de bu şekilde yapılabilir.

Eğer sonuçların gösterildiği bu sayfa da stil değişikliği yapmak isterseniz proje klasörünüzdeki **templates/admin/change\_list\_results.html** dosyasını kullanabilirsiniz.

Örnek stil:

```
<style>
.object-tools a.addlink {
    font-size: 22px;
    background-color: #ff9900;
}
</style>
```

TEKNİSYEN EKLE+

SÜZ

Aktif süzgecine göre

Tümü  
Evet  
Hayır

Adı Soyadı süzgecine göre

Tümü  
Cem Emir YÜKSEKTEPE

www.  
Bağış H

– 2016  
turkiye.com  
00 0006 6675 10

Yine aynı dosyada bulunan sonuç sayısını yazdırabiliriz.

....

...

```
{% if results %}
<div class="results">

<p>{{ results|length }} adet kayıt bulundu!</p>

<table id="result_list">
```

.....

..

.

Değiştirmek için Teknisyen seçin

2016 Ocak 13

Eylem:   0 / 1 nesne seçildi

1 adet kayıt bulundu!

<input type="checkbox"/>	AKTİF	ADI SOYADI	YAZDIR	BURA!
<input type="checkbox"/>	✓	Cem Emir YÜKSEKTEPE	Yazdır	Cem_

1 Teknisyen

Şimdi de kullanıcıya göre gösterilecek alanları kısıtlamayı görelim.

list\_display'i liste olarak tanımlayarak remove ile kullanıcıya göre gizle/göster yapabiliriz.

list\_display = ['Aktif', 'AdSoyad', 'Yazdir', 'EkAlanTest']

olarak değiştirelim.

```

def has_add_permission(self, request):
    # if not request.user.is_superuser:
    KayitSayisi = self.model.objects.count()
    KalanLimit = 2

    if KayitSayisi >= KalanLimit:
        return False
    else:
        return True

def get_list_display(self, request):
    g_l = super(TeknisyenAdmin, self).get_list_display(request)

    try:
        if request.user.is_superuser:
            g_l.remove('EkAlanTest')
    except:
        pass
    return g_l

admin.site.register(Teknisyen, TeknisyenAdmin)

```

def get\_list\_display(self, request):

g\_l = super(TeknisyenAdmin, self).get\_list\_display(request)

### list\_display listesini g\_l adında bir listeye aktarıyoruz.

Try:

## kullanıcı tekrar sayfayı yüklediğinde silinen alan yeniden silinmeye çalışılacak ve hata oluşacak

if not request.user.is\_superuser:

### kullanıcı super kullanıcı değilse

g\_l.remove('EkAlanTest')

### list\_display listemizden EkAlanTest alanını gizle

except:

### hata oluştuğunda pas geç

pass

return g\_l

### g\_l listesi düzenlenmiş olarak yada normal hali ile geri gönderilsin.

Değiştirmek için Teknisyen seçin

2016 Ocak 13

Eylem:   0 / 1 nesne seçildi

1 adet kayıt bulundu!

<input type="checkbox"/>	AKTİF	ADI SOYADI	YAZDIR
<input checked="" type="checkbox"/>		Cem Emir YÜKSEKTEPE	Yazdır

1 Teknisyen

SÜZ

Aktif s

Tümü

Evet

Hayır

Adı So

Tümü

Cem Er

Alınabilecek hatalar:

### 'tuple' object has no attribute 'remove'

tuple olarak tanımlanan list\_display'i listeye çevirmeyi atladınız, yani ['Aktif', 'AdSoyad'.....] gibi olmalı

### list.remove(x): x not in list

Kullanıcı sayfaı tekrar yüklemeye çalışıldığında listeden EkAlanTest alanını tekrar silmeye çalışıyor ama yok. try except kullanmayı unuttunuz.

14 Ocak 2016

Bu fonksiyonlar tüm yetkileri kısıtlayabilirsiniz.

Şimdi de uygulamanın yönetim panelindeki isimlendirmesine bakalım.

Uygulamamızın adı **servisformu** olduğu için uygulama için oluşturulan div de SERVISFORMU olarak isimlendirildi. İstersek burayı değiştirebiliriz.

Site yönetimi

KİMLİK DOĞRULAMA VE YETKİLENDİRME	
Gruplar	<a href="#">+ Ekle</a> <a href="#">Değiştir</a>
Kullanıcılar	<a href="#">+ Ekle</a> <a href="#">Değiştir</a>
SERVISFORMU	
Teknisyenler	<a href="#">+ Ekle</a> <a href="#">Değiştir</a>

Son E
Eylemle
<a href="#">muslu</a> Kullan
<a href="#">Muslu</a> Teknis
<a href="#">Cem</a> Teknis

Bunun için \_\_init\_\_.py ve apps.py dosyalarımızı açalım.

### apps.py:

```
# -*- coding: utf-8 -*-
```

```
from __future__ import unicode_literals  
from django.apps import AppConfig
```

```
class ServisformuConfig(AppConfig):
```

```
    name = 'servisformu'
```

```
### uygulamanın adı
```

```
    verbose_name = u'Servis Formları'
```

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

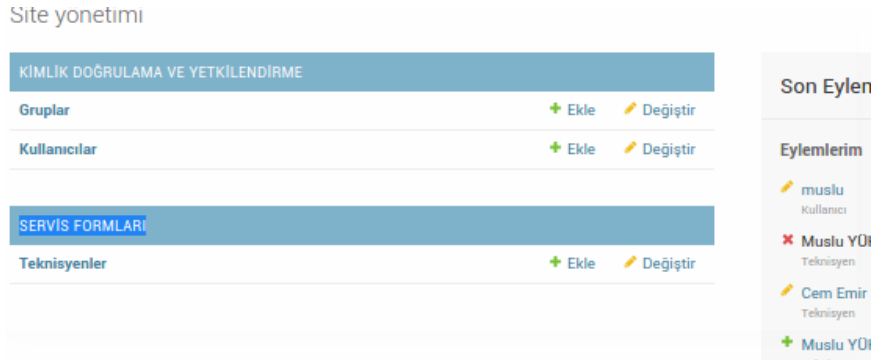
### uygulamamızın gösterilecek adı

\_\_init\_\_.py:

default\_app\_config = 'servisformu.apps.ServisformuConfig'

<http://127.0.0.1:8000/admin/>

Uygulama adımız değişti.



Projemizin konusuna geri dönersek bir teknik servis i takibi için formlar oluşturacağız. Bunun için önce projenin neleri kapsayacağı ve nelere sorun çözeceğini düşünerek ön bir fizibilete yapalım.

Çıktısı alınacak form için hangi alanlar olması gerektiğini düşünelim.

1. Müşteri bilgileri
2. Ürünler (tablet, laptop, otomobil, telefon vss.)
3. Ürünler ile gelen aksesuarlar (pil, çanta vs..)
4. Teknisyen (Müdahale edecek personel)

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

## 5. İşlem durumları (Hazır, yeni geldi, parça bekliyor vs..)

Projemizde önden kurgulamamız gereken hususlarda çok dikkatli olmalıyız. Daha sonradan eklenmesi gereken alan yada uygulamalar, veritabanında bilgi kaybına kadar yol açabilir.

Teknisyenler için bir tablo oluşturmuştuk. Şimdi de müşteriler için bir tablo oluşturalım. Bunu yeni bir uygulama oluşturarak yapabiliriz ancak ben aynı model üzerinden yeni bir sınıfla devam etmenin daha uygun olduğunu düşünüyorum.

Fazla uygulama ile yeni başlayan arkadaşların kafaları karışmasını istemeyiz.

models.py dosyamızı açıp alt tarafa yazmaya başlayalım.

models.py:

```
import random, string

class Musteriler ( models.Model ) :
    Aktif          = models.BooleanField(default = 1)
    Kodu           = models.CharField ( u'Müşteri Kodu', default=''.join(random.choice(string.digits)
for x in range(8)), max_length = 8 )
    Unvan          = models.CharField ( u'Ticari Ünvan', max_length = 250 )
    Yetkili        = models.CharField ( u'Yetkili Adı Soyadı', max_length = 250 )
    Telefon        = models.CharField ( u'Telefon', max_length = 13 )
    KayıtTarihi    = models.DateField ( u"Kayıt Tarihi", default=timezone.now)

    def __unicode__(self):
        return self.Yetkili

    class Meta:
        verbose_name_plural      = u"Müşteriler"
        verbose_name             = u"Müşteri"
```

Bu sınıfta farklı olarak kullandığımız şey müşteri koduna ait rastgele bir kod oluşturmak. Yeni bir kayıt açtığımızda default olarak sürekli rastgele 8 haneli sadece rakamlardan oluşan bir kod oluşturulacak.

```
default=''.join(random.choice(string.digits) for x in range(8))
```

terminalde

```
./manage.py makemigrations servisformu && ./manage.py migrate && ./manage.py runserver
```



ile deęiřiklikleri onaylatıp serverı bařlatıyoruz.

admin.py dosyasını aarak

```
class MusterilerAdmin(admin.ModelAdmin):  
    list_display      =  ('Kodu', 'Unvan', 'Yetkili', 'Aktif')  
    list_per_page     =  80  
    exclude          =  ('KayitTarihi', )  
    search_fields     =  ('Yetkili', 'Unvan' )
```

```
admin.site.register(MusterilerAdmin, Musteriler)
```

Alınacak hata:

**TypeError: 'MediaDefiningClass' object is not iterable**

ünkü register ederken nce sınıf, sonra sınıfa ait admin ayarları sınıfı tanımlanmalı.  
Bu hata ile ok nadir karřılacaksınız ama aklıma gelmiřken deęinmek istedim.

admin.site.register(Musteriler, MusterilerAdmin)  
olarak dzeltelim.

./manage.py makemigrations servisformu && ./manage.py migrate && ./manage.py runserver

ve

<http://127.0.0.1:8000/admin/servisformu/>

Giriř

Servis Formları

Servis Formları ynetimi

SERVIS FORMLARI

Mřteriler

+ Ekle

Deęiřtir

Teknisyenler

+ Ekle

Deęiřtir

Giriř

Servis Formları

Mřteriler

Ekle Mřteri

Mřteri ekle

☒ Aktif

Mřteri Kodu: 94506257

Ticari Unvan:

Yetkili Adı Soyadı:

Telefon:

Müşteri kodu otomatik 8 karakter ve sadece rakamlardan oluşuyor.

Labellarda gördüğünüz gibi tüm alanlar zorunlu. Çünkü charfield oluştururken blank=True kullanmadık.

models.py ye dönelim ve bir test daha yapalım.

Telefon alanımıza blank = True ekleyelim.

Telefon = models.CharField ( u'Telefon', max\_length = 13, blank = True )

Müşteri ekle

☒ Aktif

Müşteri Kodu:	<input type="text" value="18572984"/>
Ticari Ünvan:	<input type="text"/>
Yetkili Adı Soyadı:	<input type="text"/>
Telefon:	<input type="text"/>

Artık telefon alanı zorunlu değil.

Bonus: Gerektiğinde bir alanı **null** kabul edilmiyor istersenirse **null = True** yazabilirsiniz.

✓ Müşteri "Muslu YÜKSEKTEPE" başanlı olarak eklendi.

Değiştirmek için Müşteri seçin MÜŞTERİ EKLE+

Q

Eylem:   0 / 1 nesne seçildi

1 adet kayıt bulundu!

<input type="checkbox"/> MÜŞTERİ KODU	TİCARİ ÜNVAN	YETKİLİ ADI SOYADI	AKTİF
<input checked="" type="checkbox"/> 18572984	Yazki Bilişim Hizmetleri	Muslu YÜKSEKTEPE	<input checked="" type="checkbox"/>

1 Müşteri

NOT: Kayıt düzenleme linki her zaman için satırın ilk alanındadır. Burada müşteri kodu.

Satırdaki tüm alanlarda yada seçilen alanlar tıklandığında düzenleme linkinin getirilmesi için admin.py dosyamızı açıp `list_display_links` ekleyebiliriz.

```
class MusterilerAdmin(admin.ModelAdmin):
    list_display = ('Kodu', 'Unvan', 'Yetkili', 'Aktif')
    list_per_page = 80
    exclude = ('KayitTarihi',)
    search_fields = ('Yetkili', 'Unvan')
    list_display_links = ('Unvan', 'Yetkili')
```

Eylem:   0 / 1 nesne seçildi

1 adet kayıt bulundu!

<input type="checkbox"/>	MÜŞTERİ KODU	TİCARİ UNVAN	YETKİLİ ADI SOYADI	AKTİF
<input type="checkbox"/>	18572984	Yazki Bilişim Hizmetleri	Muslu YÜKSEKTEPE	<input checked="" type="checkbox"/>

1 Müşteri

Kayıtlarda toplu işlem yaptırmak için action ları kullanırız. Türkçe karşılığı eylem olarak kullanılmış ve bir açılır kutuda sunulmuş. İstersek bunlara kendimize özel toplu işlem eylemleri ekleyebiliriz.

Eylem:   1 / 1 seçildi

1 adet k

Seçili Müşteriler nesnelerini sil

<input checked="" type="checkbox"/>	MÜŞTERİ KODU	TİCARİ UNVAN
<input checked="" type="checkbox"/>	18572984	Yazki Bilişim Hizmetleri

1 Müşteri

admin.py dosyamızı açalım ve en üstlerden bir yer beğenip aşağıdaki fonksiyonu ekleyelim.

```
def SecilileriGuncelle(modeladmin, request, queryset):
    for k in queryset:
        k.save()
    return ""
SecilileriGuncelle.short_description = u"Seçilileri Güncelle"
```

```
class MusterilerAdmin(admin.ModelAdmin):
    list_display = ('Kodu', 'Unvan', 'Yetkili', 'Aktif')
```

```
list_per_page = 80
exclude = ('KayitTarihi',)
search_fields = ('Yetkili', 'Unvan')
list_display_links = ('Unvan', 'Yetkili')
actions = (SecilileriGuncelle,)
actions_on_bottom = True
actions_on_top = True
```

```
def SecilileriGuncelle(modeladmin, request, queryset):
    for k in queryset:
        k.save()
    return ""
SecilileriGuncelle.short_description = u"Secilileri Guncelle"

class MusterilerAdmin(admin.ModelAdmin):
    list_display = ('Kodu', 'Unvan', 'Yetkili', 'Aktif')
    list_per_page = 80
    exclude = ('KayitTarihi',)
    search_fields = ('Yetkili', 'Unvan')
    list_display_links = ('Unvan', 'Yetkili')
    actions = (SecilileriGuncelle,)
    actions_on_bottom = True
    actions_on_top = True
```

Bonus: **actions\_on\_bottom** ve **actions\_on\_top** ile eylemler açılır kutusunu kayıt gridlerinin altında yada üstünde göster diyebiliriz.

Alınabilecek hatalar:

**NameError: name 'SecilileriGuncelle' is not defined**  
**SecilileriGuncelle** fonksiyonunu altta bıraktınız yada yazmadınız.

Değiştirmek için Müşteri seçin MÜŞTERİ EKLE+

Q

Eylem:   0 / 1 nesne seçildi

1 adet k

☐ Seçili Müşteriler nesnelerini sil  
☐ Seçilileri Güncelle

MÜ	TİCARİ UNVAN	YETKİLİ ADI SOYADI	AKTİF
<input type="checkbox"/> 18572984	Yazki Bilişim Hizmetleri	Muslu YÜKSEKTEPE	<input checked="" type="checkbox"/>

Eylem:   0 / 1 nesne seçildi

1 Müşteri

Biraz daha detaylı inceleyelim.

**def SecilileriGuncelle(modeladmin, request, queryset):**

**### SecilileriGuncelle adında bir fonksiyon oluşturuyoruz**

```
print queryset
```

**### querysetten neler geldiğine bakıyoruz**

```
for k in queryset:
```

**### queryset ile seçilen tüm kayıtların listelerini alıyoruz.**

```
print k
```

**### kayıtları döngüden tek tek alıyoruz**

```
k.save()
```

**### sınıfımıza ait save modülünü çalıştırıyoruz. Birazdan detaylı değineceğim.**

```
return ""
```

**### geri dön ama boş dön**

SecilileriGuncelle.short\_description = u"Seçilileri Güncelle"

**#### fonksiyonun kısa bir açıklaması**

1 kayıt var

```
January 14, 2016 - 11:47:18
Django version 1.9.1, using settings 'teknikservistakibi.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[<Musteriler: Muslu YÜKSEKTEPE>]
Muslu YÜKSEKTEPE
[14/Jan/2016 11:47:26] "POST /admin/servisformu/musteriler/ HTTP/1.1" 302 0
[14/Jan/2016 11:47:26] "GET /admin/servisformu/musteriler/ HTTP/1.1" 200 7495
[14/Jan/2016 11:47:26] "GET /admin/jsi18n/ HTTP/1.1" 200 7259
```

2 kayıt var.

```
[14/Jan/2016 11:48:06] "POST /admin/servisformu/musteriler/add/ HTTP/1.1" 302 0
[14/Jan/2016 11:48:06] "GET /admin/servisformu/musteriler/ HTTP/1.1" 200 8143
[14/Jan/2016 11:48:06] "GET /admin/jsi18n/ HTTP/1.1" 200 7259
[<Musteriler: Serkan Eminç>, <Musteriler: Muslu YÜKSEKTEPE>]
Serkan Eminç
Muslu YÜKSEKTEPE
[14/Jan/2016 11:48:10] "POST /admin/servisformu/musteriler/ HTTP/1.1" 302 0
[14/Jan/2016 11:48:10] "GET /admin/servisformu/musteriler/ HTTP/1.1" 200 7988
```

Hatırlarsanız `__unicode__` fonksiyonuna sadece yetkili alanımızı eklemiştik.

```
def __unicode__(self):
    return self.Yetkili
```

yerine

```
def __unicode__(self):
```

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

```
# return self.Kodu + " " + self.Yetkili
# return "%s %s" % (self.Kodu, self.Yetkili)
return "Müşteri Kodu: %s - Adı: %s" % (self.Kodu, self.Yetkili)
```

Değiştirip çıktıları tekrar kontrol edelim.

```
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[<Musteriler: 69837932 Serkan Eminç>, <Musteriler: 18572984 Muslu YÜKSEKTEPE>]
69837932 Serkan Eminç
18572984 Muslu YÜKSEKTEPE
[14/Jan/2016 11:52:51] "POST /admin/servisformu/musteriler/ HTTP/1.1" 302 0
```

```
[14/Jan/2016 11:55:53] "GET /static/admin/img/toolbar-add.svg HTTP/1.1" 304 0
[<Musteriler: Müşteri Kodu: 69837932 - Adı: Serkan Eminç>, <Musteriler: Müşteri Kodu: 18572984 - Adı: Muslu YÜKSEKTEPE>]
Müşteri Kodu: 69837932 - Adı: Serkan Eminç
Müşteri Kodu: 18572984 - Adı: Muslu YÜKSEKTEPE
```

Alınabilecek hatalar:

**coercing to Unicode: need string or buffer, datetime.date found**

dönüşlerin mutlaka string olarak gönderilmesi gerekir. Burada datetimefield kullanılan bir alanı direk göndermeye çalışıyoruz.

Hazır güncelleme yaparken seçilen kayıtlarda değişim yaparak kayıt edilmesine bakalım.

admin.py dosyamızı tekrar açıp

```
def SecilileriGuncelle(modeladmin, request, queryset):
```

```
    for k in queryset:
```

```
        k.Yetkili = k.Yetkili + " ____"
```

```
        k.save()
```

```
    return ""
```

```
SecilileriGuncelle.short_description = u"Seçilileri Güncelle"
```

olarak değiştirip kayıt edelim ve test edelim.

İstediğiniz kayıtları seçip eylemlerden Seçilileri Güncelle 'yi seçip git diyelim.

2 adet kayıt bulundu!

<input type="checkbox"/>	MÜŞTERİ KODU	TİCARİ UNVAN	YETKİLİ ADI SOYADI	AKTİF
<input type="checkbox"/>	69837932	Esefix	Serkan Eminç____	✓
<input type="checkbox"/>	18572984	Yazki Bilişim Hizmetleri	Muslu YÜKSEKTEPE____	✓

Eylem:   0 / 2 nesne seçildi

Yetkili adı soyadı alanlarındaki kayıtların sonuna \_\_\_\_ eklendi.

Buraya da özel bir alan ekleyerek müşterilerin telefon numaralarını aratabiliriz. Bunu için

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

tel:.... kullanacağız yani bilgisayarınızda arama yapan bir uygulamaya yada sanal santraliniz varsa bir apiye ihtiyacınız olacak.

admin.py:

```
class MusterilerAdmin(admin.ModelAdmin):
    list_display = ('Kodu', 'Unvan', 'Yetkili', 'Aktif', 'AramaYap')
    ...
    ...
```

models.py:

```
class Musteriler ( models.Model ) :
    ..
    def AramaYap ( self ) :
        if self.Telefon:
            return '<a href="tel:%s" target="_blank">Numarayı Ara</a>' % self.Telefon
        else:
            return 'Telefon No kayıt edilmedi'
    AramaYap.short_description = u'Ara'
    AramaYap.allow_tags = True
```

olarak değiştirelim.

Eylem:   0 / 2 nesne seçildi

2 adet kayıt bulundu!

<input type="checkbox"/>	MÜŞTERİ KODU	TİCARİ UNVAN	YETKİLİ ADI SOYADI	AKTİF	ARA
<input type="checkbox"/>	69837932	Esefix	Serkan Eminç____	✓	Telefon No kayıt edilmedi
<input type="checkbox"/>	18572984	Yazki Bilişim Hizmetleri	Muslu YÜKSEKTEPE____	✓	Numarayı Ara

Eylem:   0 / 2 nesne seçildi

Güncelleme yaparken sql karşılığını da görmek isterseniz.

admin.py

```
def SecilileriGuncelle(modeladmin, request, queryset):
    print queryset.query
    for k in queryset:
        k.Yetkili = k.Yetkili + " ____"
        k.save()
    return ""
SecilileriGuncelle.short_description = u"Seçilileri Güncelle"
```

olarak fonksiyonu güncelleyip terminal çıktılarına bakabiliriz.

```
Terminal
+ Starting development server at http://127.0.0.1:8000/
X Quit the server with CONTROL-C.
SELECT "servisformu_musteriler"."id", "servisformu_musteriler"."Aktif", "servisformu_musteriler"."Kodu", "servisformu_musteriler"."Unvan", "servisformu_musteriler"."Yetkili", "servisformu_musteriler"."Telefon", "servisformu_musteriler"."KayitTarihi" FROM "servisformu_musteriler" WHERE "servisformu_musteriler"."id" IN (2, 1) ORDER BY "servisformu_musteriler"."id" DESC
[14/Jan/2016 12:26:12] "POST /admin/servisformu/musteriler/ HTTP/1.1" 302 0
```

Şimdide bazı koşulları sağlayan kayıtlarda toplu güncelleme yapalım.

admin.py

```
def SecilileriGuncelle(modeladmin, request, queryset):
    print queryset.query
    for k in queryset:
        if k.Yetkili.startswith("Muslu"):
            k.Yetkili = k.Yetkili + u"__Musluilebaşlıyordu"
            k.save()
    return ""
SecilileriGuncelle.short_description = u"Seçilileri Güncelle"
```

Sizde super kullanıcı ise ve telefon numaralarının başına 0 eklenmediyse 0 ekleyerek yeniden kaydet olarak deneme yapabilirsiniz.

Eylem:   0 / 2 nesne seçildi

2 adet kayıt bulundu!

<input type="checkbox"/>	MÜŞTERİ KODU	TİCARİ UNVAN	YETKİLİ ADI SOYADI	AKTİF	ARA
<input type="checkbox"/>	69837932	Esefix	Serkan Eminç	<input checked="" type="checkbox"/>	Telefon No kayıt edilmedi
<input type="checkbox"/>	18572984	Yazki Bilişim Hizmetleri	Muslu YÜKSEKTEPE__Musluilebaşlıyordu	<input checked="" type="checkbox"/>	Numarayı Ara

Eylem:   0 / 2 nesne seçildi

Alınabilecek hatalar:

**'ascii' codec can't decode byte 0xc5 in position 13: ordinal not in range(128)**

Türkçe karakter kullandıysanız ama utf-8 kullanmanız gerekiyorsa bu hatayı alırsınız. Örnek olarak "\_\_Musluilebaşlıyordu" başına u"...." eklenmediyse bu hata alınır.

Sanırım özel alan ekleyip işlem yaptıрма daha iyi anlaşılmıştır.

Bu şekilde bir çok özellik ve güzellik ekleyebilirsiniz.

Teknisyen ve müşterilerin kayıtlarını yapabildiğimize göre artık diğer gereksinimleri yazabiliriz.



Alınan ürünün hangi durumda olduğunu kayıt edip, her işlemde güncelleyip ve sorguda göstermek için durumları hazırlamamız gerekiyor.

Sabit durumlar kullanılacak ise charfield e choices ekleyerek bir listeden seçim yaptırabiliriz ama biz siteyi kullanacak kişilerin kendi durumlarını ekleyebilmeleri için yeni bir tablo oluşturacağız.

Models.py dosyamızı açalım ve Durumlar adında bir sınıf oluşturalım

models.py:

```
..  
..  
class Durumlar (models.Model):  
    Durumu      = models.CharField(u'Durum', max_length=30, help_text='Metin alanının altında  
    kayıt girerken yardımcı olabilecek konuları anlatan kısa bir açıklama yazabilirsiniz.')
```

```
    def __unicode__(self):  
        return self.Durumu
```

```
    class Meta:  
        verbose_name_plural      = u"Durumlar"  
        verbose_name             = u"Durum"
```

help\_text kullanımına dikkat edin.

admin.py:

```
..  
class DurumlarAdmin(admin.ModelAdmin):  
    list_display      = ('Durumu',)  
    list_per_page     = 80
```

admin.site.register(Durumlar, DurumlarAdmin)

Terminalde;

**./manage.py makemigrations servisformu && ./manage.py migrate && ./manage.py runserver**

Giriş > Servis Formları		
Servis Formları yönetimi		
SERVIS FORMLARI		
Durumlar	<a href="#">+ Ekle</a>	<a href="#">Değiştir</a>
Müşteriler	<a href="#">+ Ekle</a>	<a href="#">Değiştir</a>
Teknisyenler	<a href="#">+ Ekle</a>	<a href="#">Değiştir</a>

Durum ekle

Durum:

Metin alanının altında kayıt girerken yardımcı olabilecek konulan anlatan kısa bir açıklama yazabilirsiniz.

Kaydet ve başka birini ekle

Kaydet ve düzenlemeye devam et

KAYDET

Bonus: Yukarıdaki Giriş > Servis Formları > Durumlar > Ekle Durum yazısı kafanıza takıldı ve Durum Ekle olarak değiştirmek isterseniz ( tüm kayıtlarda geçerli olacak ) **template/admin/change\_form.html** dosyasını açın, ctrl+g 21 ( farklı satır olabilir ) yazarak if add koşulunu bulunup aşağıdaki gibi değiştirin.

{% trans 'Add' %} etiketini yer değiştirin. Translate konusuna sonra gireceğiz.

```
&rsquo; {% if add %} {{ opts.verbose_name }} {% trans 'Add' %} {% else %} {{ original|truncatewords:"18" }}
{% endif %}
```

Durum ekle

Durum:

Metin alanının altında kayıt girerken yardımcı olabilecek konulan anlatan kısa bir açıklama yazabilirsiniz.

Kaydet ve ba

Servise bırakılan ürünlerin durumlarını da ayarladığımıza göre ürünle beraber bırakılan aksesuarları da kayıt altına alalım.

models.py:

...

...

```
class Aksesuarlar (models.Model):
```

```
    Adi = models.CharField(u'Adi', max_length=30, help_text='Ürünle beraber getirilen tüm aksesuarlar. Örn: Batarya, Çanta')
```

```
    def __unicode__(self):
        return self.Adi
```

```
class Meta:
    verbose_name_plural = u"Aksesuarlar"
    verbose_name = u"Aksesuar"
```

admin.py:

...  
..  
.

```
class AksesuarlarAdmin(admin.ModelAdmin):  
    list_display      = ('Adi',)  
    list_per_page     = 5
```

admin.site.register(Aksesuarlar, AksesuarlarAdmin)

./manage.py makemigrations servisformu && ./manage.py migrate && ./manage.py runserver

SERVIS FORMLARI	
Aksesuarlar	<a href="#">+ Ekle</a> <a href="#">Değiştir</a>
Durumlar	<a href="#">+ Ekle</a> <a href="#">Değiştir</a>
Müşteriler	<a href="#">+ Ekle</a> <a href="#">Değiştir</a>
Teknisyenler	<a href="#">+ Ekle</a> <a href="#">Değiştir</a>

Giriş · Servis Formları · Aksesuarlar · Aksesuar Ekle

Aksesuar ekle

Adi:

Çanta

Ürünle beraber getirilen tüm aksesuarlar. Örn: Batarya, Çanta

Kaydet ve başka birini ekle

Kaydet ve düzenlemeye devam et

KAYDET

Aksesurları da eklediğimize göre artık servis formumuzu oluşturabiliriz. Her forma bir ürün eklenecekse önce ürün sınıfını hazırlamamız gerekiyor ama bir form da getirilen tüm ürünlerin kaydı tutulması istenirse önce formu hazırlamalıyız ki inline ile ürünleri sınırsız kayıt ettirebileyim. Kafanız karışmasın hemen yazmaya başlıyoruz.

models.py:

...  
.

```
class ServisForm ( models.Model ) :  
    Musteri      = models.ForeignKey ( Musteriler )  
    TeslimEden   = models.CharField ( u'Teslim Eden', max_length = 130 )  
    TeslimAlan   = models.ForeignKey ( Teknisyen, default=int(Teknisyen.objects.get(id=1).id) )  
    FormNo       = models.CharField ( u'Form No', default=''.join(random.choice(string.digits) for x in  
range(8)), max_length = 8 )  
    KayitTarihi  = models.DateTimeField ( u"Kayıt Tarihi", default=timezone.now)
```

```
def __unicode__(self):  
    return self.FormNo
```

class Meta:

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

```
verbose_name_plural = u"Formlar"  
verbose_name = u"Servis Formu"
```

```
def Yazdir ( self ) :  
    return '<a href="/yazdir/%s" target="_blank">Yazdır</a>' % self.id  
Yazdir.short_description = u'Yazdır'  
Yazdir.allow_tags = True
```

Not:

ForeignKey ile başka bir tabloya indexleme yaptırıyoruz. NoSQL kullanacağımız zaman bunu kullanamayız. TeslimAlan alanına default olarak teknisyen tablosundaki ilk kaydı getirteceğiz.

Alınabilecek hatalar:

**This query requires pytz, but it isn't installed.**

Çözümü: terminalde **sudo pip install pytz**

admin.py:

```
class ServisFormAdmin(admin.ModelAdmin):  
    list_display = ( 'FormNo', 'Musteri', 'TeslimEden', 'TeslimAlan', 'KayitTarihi', 'Yazdir' )  
    list_per_page = 50  
    ordering = ( '-KayitTarihi', )  
    date_hierarchy = 'KayitTarihi'  
    search_fields = ( 'FormNo', 'Musteri__Yetkili', 'Musteri__Telefon' )  
    exclude = ( 'KayitTarihi', )
```

admin.site.register(ServisForm, ServisFormAdmin)




search\_fields tanımlamasına dikkat ettiyseniz “\_\_” kullandık. Bu Musteri tablosundaki Yetkili ve Telefon alanlarında da ara demek.

**./manage.py makemigrations servisformu && ./manage.py migrate && ./manage.py runserver**




**Yeni kayıt oluşturmayın. Ürünleri ekledikten sonra oluşturacağız.**

Giriş > Servis Formları > Formlar > Servis Formu Ekle

Servis Formu ekle

Musteri:    

Teslim Eden:

Teslim Alan:    

Form No:

Dağıtım Hesabı: 1100 0000 2000 7000 0000 0070 10

Hatırlarsanız müşterilere ve teslim alan personel için aktif kısıtlaması eklemiştik. Bu kısıtı kullanmak için admin.py dosyamızı açalım.

```
class ServisFormAdmin(admin.ModelAdmin):
    list_display = ('FormNo', 'Musteri', 'TeslimEden', 'TeslimAlan', 'KayitTarihi', 'Yazdir')
    list_per_page = 50
    ordering = ('-KayitTarihi',)
    date_hierarchy = 'KayitTarihi'
    search_fields = ('FormNo', 'Musteri__Yetkili', 'Musteri__Telefon')
    exclude = ('KayitTarihi',)

    def formfield_for_foreignkey(self, db_field, request, **kwargs):
        if db_field.name == 'TeslimAlan':
            ### field ( alan ) adı TeslimAlan ise

            kwargs["queryset"] = Teknisyen.objects.filter(Aktif=True)
            ### Teknisyenlerin sadece aktif olanları filtreleyerek getir ve queryset e ekle
            ### kwargs konusuna şimdilik girmiyoruz

        if db_field.name == 'Musteri':
            kwargs["queryset"] = Musteriler.objects.filter(Aktif=True)
            return super(ServisFormAdmin, self).formfield_for_foreignkey(db_field, request, **kwargs)
```

olarak değiştirelim.

formfield\_for\_foreignkey fonksiyonu, sınıfımıza ait foreignkey fieldlarını kullanmamızı sağlar. Yani bu fieldlara kısıtlama yada yetkilendirme koyabiliriz.

Müşteri kısıtlamasını test etmek için

Giriş · Servis Formları · Müşteriler · Müşteri Kodu: 69837932 - Adı: Serkan Eminç

Müşteri değiştir

☒ Aktif

Müşteri Kodu:

69837932

Ticari Unvan:

Esefix

Yetkili Adı Soyadı:

Serkan Eminç

Telefon:

Sil

Bağış Hesabı: 1 K03 0006 2000 / 500 0006 66 / 5 10

bir müşterinin aktif durumunu pasife çekip kaydediyoruz.

Giriş · Servis Formları · Formlar · Servis Formu Ekle

Servis Formu ekle

Musteri:	<div>-----</div>
Teslim Eden:	<div>Müşteri Kodu: 18572984 - Adı: Muslu YÜKSEKTEPE_Musluilebaşlıyordu</div>
Teslim Alan:	<div>Cem Emir YÜKSEKTEPE</div>
Form No:	<div>65546242</div>

Foreignkey kısıtlamasını da böylelikle görmüş olduk.

Servis formu kaydını yapabiliyoruz ama teslim edilen ürünleri eklememiz gerekiyor. Biraz önce bahsettiğim gibi bir forma ait çok ürün ekleyebiliriz. Bu vesile ile inline tabloları da görmüş olacağız.

models.py dosyamızı açalım

Dikkat edin ServisForm dan sonra eklemeniz gerekiyor.

...  
...  
..

```
class Urunler ( models.Model ) :  
    ServisFormu      =      models.ForeignKey ( ServisForm )  
    Cins              =      models.CharField ( u'Cinsi', max_length = 30 )  
    Marka             =      models.CharField ( u'Marka', max_length = 50 )  
    Model             =      models.CharField ( u'Model', max_length = 50 )  
    SeriNo            =      models.CharField ( u'Seri No', max_length = 250 )  
    GarantiBitis      =      models.DateField ( u"Garanti Bitiş", default=timezone.now )  
    Sikayet           =      models.TextField ( u'Şikayet' )  
    Aksesuar          =      models.ManyToManyField ( Aksesuarlar, blank=True )  
    Durum             =      models.ForeignKey ( Durumlar )  
    Not               =      models.TextField ( u'Yapılan İşlemler', blank=True )
```

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

```
def __unicode__(self):
    return "%s %s %s" % (self.Cins, self.Marka, self.Model)
```

```
class Meta:
    verbose_name_plural = u"Ürünler"
    verbose_name = u"Ürün"
```

Not:

ManyToManyField kullandık, yani bir indexleme yaparak başka bir tablodaki kayıtlardan çoklu seçim yapabiliriz.

TextField kullandık, Uzun açıklamalar girebiliriz.

admin.py dosyamızı açalım

admin.py:

```
..
..
.
```

```
class UrunlerInline(admin.StackedInline):
```

```
    model = Urunler
```

```
###hangi model
```

```
    extra = 0
```

**### yeni eklenmek istendiğinde kaç tane ürün ekleme yapsın. İlk kayıta hazır olarak ürün ekleme sayfası getirmeyecek biz ekle deyince 1 tane ürün ekleme sayfası açacak.**

```
    max_num = 5
```

```
### en fazla kayıt ürün eklenebilir.
```

```
class ServisFormAdmin(admin.ModelAdmin):
```

```
    inlines = [ UrunlerInline, ]
```

```
    list_display = ( 'FormNo', 'Musteri', 'TeslimEden', 'TeslimAlan', 'KayitTarihi', 'Yazdir' )
```

```
    list_per_page = 50
```

```
    ordering = ( '-KayitTarihi', )
```

```
    date_hierarchy = 'KayitTarihi'
```

```
    search_fields = ( 'FormNo', 'Musteri__Yetkili', 'Musteri__Telefon' )
```

```
    exclude = ( 'KayitTarihi', )
```

```
def formfield_for_foreignkey(self, db_field, request, **kwargs):
```

```
    if db_field.name == 'TeslimAlan':
```

```
        kwargs["queryset"] = Teknisyen.objects.filter(Aktif=True)
```

```
    if db_field.name == 'Musteri':
```

```
kwargs["queryset"] = Musteriler.objects.filter(Aktif=True)

return super(ServisFormAdmin, self).formfield_for_foreignkey(db_field, request, **kwargs)
```



Not:

UrunlerInline sınıfını register etmiyoruz.



**./manage.py makemigrations servisformu && ./manage.py migrate && ./manage.py runserver**

Giriş · Servis Formları · Formlar · Servis Formu Ekle

### Servis Formu ekle


Musteri:   

Teslim Eden:

Teslim Alan:   

Form No:

### ÜRÜNLER

 Başka bir Ürün ekle

extra=0 dediğimiz için ürün sayfası açık gelmedi, **Başka bir Ürün ekle** linki ile ürün ekleme sayfamız gelecek.



## Not:

Tasarımsal olarak daha uygun olduğunu StackedInline kullandık.

TeslimAlan:	Cem Emir YÜKSEKTEPE	✎ +
Form No:	87814917	
<b>ÜRÜNLER</b>		
Ürün: #1		
Cinsi:		
Marka:		
Model:		
Seri No:		
Garanti Bitiş:	2016-01-14	Bugün   📅
Şikayet:		
Aksesuar:	<div>Çanta Adaptör Batarya</div> +	
Birden fazla seçmek için "Control (Ctrl)" veya Mac'deki "Command" tuşuna basılı tutun.		

TabularInline kullandaysık yana doğru uzayacaktı.

ÜRÜNLER					
CİNSİ	MARKA	MODEL	SERİ NO	GARANTİ BİTİŞ	\$
				2016-01-14 Bugün   📅	
+ Başka bir ürün ekle					
			Kaydet ve başka birini ekle	Kaydet ve düzenlemeye devam et	KAYDET

Servis formumuza yeni bir kayıt girdiğimizde aşağıdaki gibi gözükecek.

✓ Servis Formu "66024779" başarılı olarak eklendi.

Değiştirmek için Servis Formu seçin

SERVİS FORMU EKLE+

2016 Ocak 14

Eylem:   0 / 1 nesne seçildi

1 adet kayıt bulundu!

FORM NO	MÜSTERİ	TESLİM EDEN	TESLİMALAN	KAYIT TARİHİ	YAZDIR
66024779	Müşteri Kodu: 11278334 - Adı: Muslu YÜKSEKTEPE	Ceylan YÜKSEKTEPE	İsmail İSİRGAN	Oca. 14, 2016, 3:27 ö.s.	<a href="#">Yazdır</a>

1 Servis Formu

**Bonus:** Veritabanı işlemleri için ben RazorSQL kullanıyorum. Bilinen tüm veritabanlarını destekliyor ve kullanımı çok basit.

<http://www.razorsql.com>

Oluşturduğumuz servis formunun çıktısını almak için yazdır adında bir fonksiyon oluşturup grid e eklemiştik. Şimdi de bu kaydın html üzerinde gösterimine değinelim.

views.py dosyamızı açıp yazdıracağımız kaydın id si ile eşleştirme yaparak elde edilen değerleri html dosyasına göndereceğiz.

views.py:

```
def sayfaiyazdir(request, idsi):
```

```
### request şart. Olmazsa olmaz. Bu fonksiyona en azından bir id değeri gerekiyor.
```

```
    formdurumu = ServisForm.objects.get(id = idsi)
```

```
## bu id ye ait form bilgileri
```

```
    formbilgileri = Urunler.objects.filter(ServisFormu__id = idsi)
```

```
### buid ye ait ürünlerin listesi
```

```
#### ServisFormu__id ile gelen id yi indexlediğimiz servisformu tablosundaki id ile eşleştiriyoruz.
```

```
### örnek olması için bu şekilde yazdım.
```

Muslu YÜKSEKTEPE – 2016

[www.muslu.org](http://www.muslu.org) | [www.djangoturkiye.com](http://www.djangoturkiye.com)

Bağış Hesabı: TR03 0006 2000 7500 0006 6675 10

```
return render(request, 'yazdir.html', {'formbilgileri': formbilgileri, 'formdurumu': formdurumu})  
### bulduğumuz bilgileri yazdir.html dosyasına gödneriyoruz. Tabi ki templates klasörümüzde
```

urls.py dosyamızı açalım ve kodlarımızı yazalım.

```
from servisformu.views import sayfaiyazdir
```

```
urlpatterns = [
```

```
    url(r'^yazdir/([\w\-\+]+)/$', sayfaiyazdir, name='sayfaiyazdir'),  
    url(r'^admin/', admin.site.urls),
```

```
]
```

/yazdir/\*\*\*/ diye bir link geldiğinde servisformu uygulama klasörümüzdeki views.py dosyasında ki sayfaiyazdir fonksiyonunu çalıştıracaktır.

Not:

([\w\-\+]) herşeyi yazabiliriz. (+id...) kullanabilirdik ama bu daha çok işinize yarayacak, aklınızda bulunsun.

Templates klasörümüzde yazdir.html adında bir dosya oluşturalım.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <!--[if IE]><meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"><![endif]-->  
    <meta charset="utf-8">  
    <title>İzmir Teknik Servis, Servis Formu kontrol, durum öğrenme | izmirteknikservis.tk</title>  
    <meta name="description" content="İzmir Teknik Servis, Servis Formu kontrol, durum öğrenme"/>  
    <link rel="canonical" href="http://www.izmirteknikservis.tk"/>  
    <link href='http://fonts.googleapis.com/css?family=Ubuntu+Mono' rel='stylesheet' type='text/css'>  
  
    <link rel="shortcut icon" href="/media/favicon.ico">  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">  
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>  
    <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>  
    <style>  
        .borderless tbody tr td, .borderless tbody tr th, .borderless thead tr th {  
            border: none;  
        }  
    </style>  
</head>
```

```
<body onload="window.print()">
```

```
<div class="container">
```

```
  <h1>{{ formdurumu.FormNo }}</h1>
```

```
  <p>{{ formdurumu.Musteri.Unvan|title }} {{ formdurumu.Musteri.Yetkili|title }} -  
  {{ formdurumu.KayitTarihi }}</p>
```

```
  <table class="table table-condensed">
```

```
    <thead>
```

```
      <tr>
```

```
        <th>Ürün</th>
```

```
        <th>Seri No</th>
```

```
        <th>Şikayet</th>
```

```
        <th>Aksesuar</th>
```

```
        <th>Not</th>
```

```
      </tr>
```

```
    </thead>
```

```
    <tbody>
```

```
      {% for k in formbilgileri %}
```

```
        <tr>
```

```
          <td>{{ k.Cins|upper }} | {{ k.Marka|upper }} | {{ k.Model|upper }}</td>
```

```
          <td>{{ k.SeriNo|upper }}</td>
```

```
          <td>{{ k.Sikayet|title }}</td>
```

```
          <td>{% for kk in k.Aksesuar.all %}{{ kk|title }} {% endfor %}</td>
```

```
          <td>{{ k.Not }}</td>
```

```
        </tr>
```

```
      {% endfor %}
```

```
    </tbody>
```

```
</table>
```

```
</div>
```

```
<div class="container">
```

```
  <table class="table table-condensed borderless">
```

```
    <tr>
```

```
      <th>Servise Teslim Eden</th>
```

```
      <th>Teknisyen</th>
```

```
      <th>Ürünü Teslim Alan</th>
```

```
    </tr>
```

```
    <tr>
```

```
      <th>{{ formdurumu.TeslimEden|title }}</th>
```

```
      <th>{{ formdurumu.TeslimAlan|title }}</th>
```

```
      <th></th>
```

```
    </tr>
```

```
</table>
```

</div>

<div class="container">

<table border="0" cellpadding="" width="100%">

<tr><td><i style="font-size:12px;">Bakım ve onarım süresi 3 (üç) aydır.</i></td></tr>

<tr><td><i style="font-size:12px;">Servisimizde asla kaçak yazılım kullanılmamakta ve destek verilmemektedir.</i></td></tr>

<tr><td><i style="font-size:12px;">Tamire gelen cihazların harddisk, yazılım ve yedeklerinden tarafımız sorumlu değildir.</i></td></tr>

<tr><td><i style="font-size:12px;">Servise gelen ürünlerde önceden bildirilmeyen sorunlardan tarafımız sorumlu değildir.</i></td></tr>

<tr><td><i style="font-size:12px;">Servise teslim edilen ürünlerin emanet süresi 30 (otuz) gün olup, sonrasında firmamızın sorumluluğu yoktur.</i></td></tr>

<tr><td><i style="font-size:12px;">Servise gelen cihazların chip değişim ve kalıplanmasında oluşabilecek sorunlarda tarafımız sorumlu değildir.</i></td></tr>

</table>

</div>

</body>

</html>

Şimdi servis formuna kayıt girerek yazdır linkini tıklayabiliriz.

Örn:

<http://127.0.0.1:8000/yazdir/1/>

Aşağıda ilk kaydımıza ait çıktıyı görebiliriz.

Bonus:

Foreign ve ManyToMany gibi indexleme durumlarında gelen elementlerin yanında düzenlemek ve yeni kayıt eklemek için simgeler yardımcı olacaktır.

Herhangi bir kaydı seçtiğinizde düzenleme aktif olur. Ek pencerede yeni kayıt yada düzenleme işlemi yapabilirsiniz.

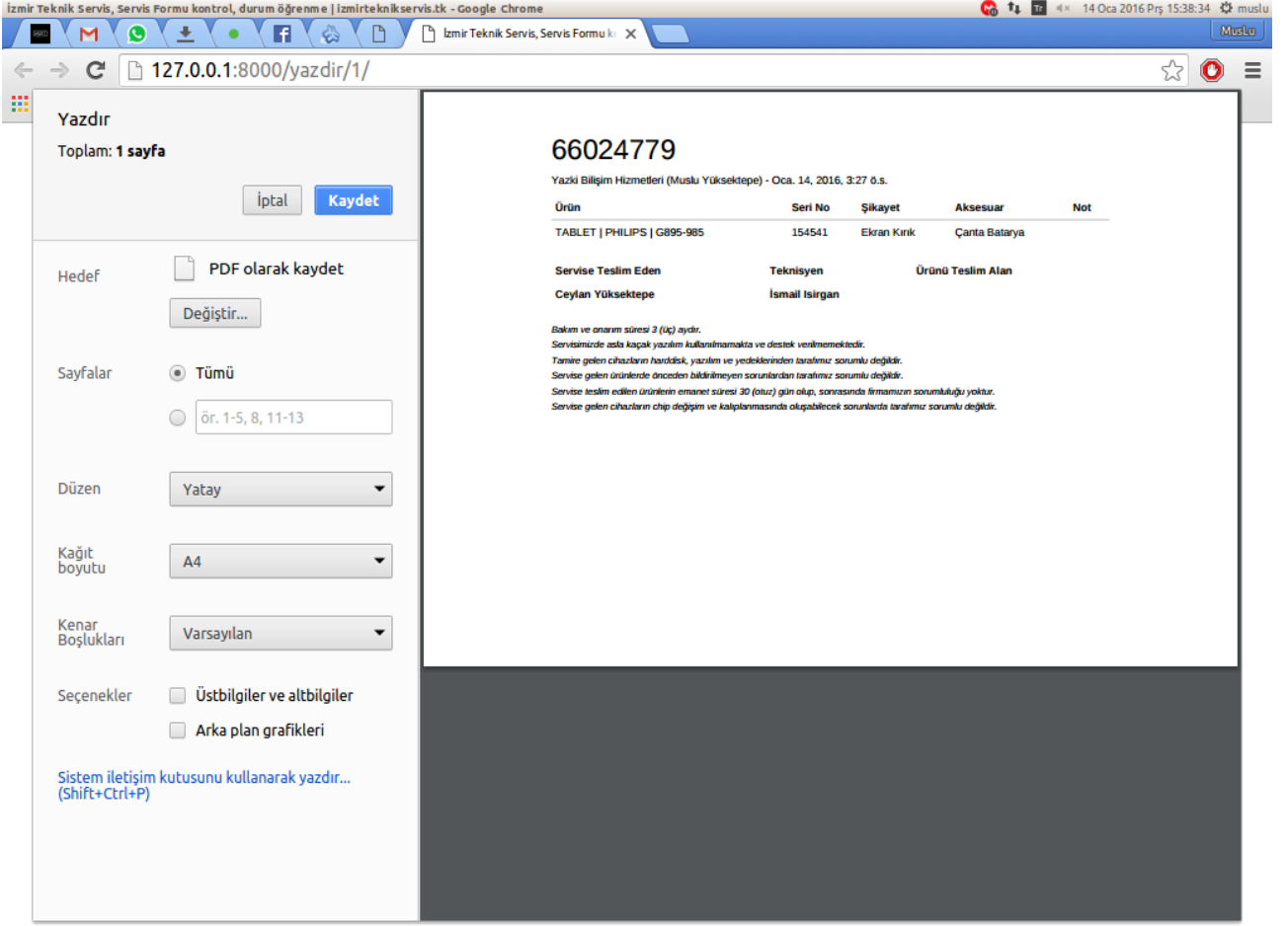
Musteri:

Müşteri Kodu: 11278334 - Adı: Muslu YÜKSEKTEPE ▼



Teslim Eden:

-----



Aşağı yukarı bir serviste olması gereken tüm eksiklikleri giderin bir web sitesi hazırlamış olduk. Kayıt girerek eksikleri görebilir test edebilirsiniz.

Ve geri bildirim olarak dönerseniz sevinirim.

Yarın daha fazla admin sayfası düzenleme ve models.py da save modüllerine bakacağız ve hata alarak çalışmalara devam edeceğiz.

Muslu YÜKSEKTEPE  
14.01.2016

15 Ocak 2016

Dün admin sayfasından gelen yazdır linkine cevap olarak sorgulama yaptırıp direk yazdır komutu vermiştik.

Bugün de views.py ile veri çekip ekrana bastırmaya ve veri sorgulamaya bakacağız.

Önce direk html dosyasına yönlendirme yaparak bu konuya da değinmiş olalım.

urls.py dosyamıza aşağıdaki kodları yazalım.

```
import edilecek fonksiyon
from django.views.generic import TemplateView
```

TemplateView hazır fonksiyondur ve view gibi kullanılarak direk html (txt, xml de olabilir) dosyasına yönlendirme yapabiliriz.

```
url(r'^$', TemplateView.as_view(template_name="index.html")),
```

bu fonksiyonu kullanabilmemiz için templates klasöründe index.html dosyası oluşturmuş olmamız gerekiyor.

index.html:

```
<!DOCTYPE HTML>
<html>
<head>
  <!--[if IE]><meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"><![endif]-->
  <meta charset="utf-8">
  <title>İzmir Teknik Servis, Servis Formu kontrol, durum öğrenme | izmirteknikservis.tk</title>
  <meta name="description" content="İzmir Teknik Servis, Servis Formu kontrol, durum öğrenme"/>
  <link rel="canonical" href="http://www.izmirteknikservis.tk"/>
  <link href='http://fonts.googleapis.com/css?family=Ubuntu+Mono' rel='stylesheet' type='text/css'>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="shortcut icon" href="/media/favicon.ico">
  <link rel="stylesheet" href="/media/styles.css">

</head>
<body>
<div id='wrap'>
  <div id="sonuclarimuslu"
    style="color:#ffffff;font-size:2.4vw;position:fixed;width:95%;top:20%;left:50%;-webkit-
transform:translate(-50%, -50%);transform: translate(-50%, -50%);">
```

```

</div>
<div id='content'>
  <section class='input'>
    <div>
      <input type='text' placeholder='Form No' id="arama" maxlength="8">
    </div>
  </section>
</div>
</div>
<script src="/media/html5shiv.min.js"></script>
<script src="/media/jquery-1.9.1.min.js"></script>
<script src="/media/fancyInput.js"></script>

<script>

  $('section :input').val('').fancyInput()[0].focus();

  var aramam = document.getElementById('arama');
  var sonuclarimuslu = document.getElementById('sonuclarimuslu');

  aramam.addEventListener('keypress', function (event) {

    if (event.keyCode == 13) {

      sonuclarimuslu.innerHTML = "";
      if (aramam.value.length == 8) {

        sonuclarimuslu.innerHTML = "Sorgulanıyor...<br /> Lütfen bekleyiniz.";
        $.post("/kn/", {
          TakipNo: aramam.value,
          csrfmiddlewaretoken: "{{ csrf_token }}"
        }, function (data) {
          sonuclarimuslu.innerHTML = data;
        });

      } else {
        sonuclarimuslu.innerHTML = "Uygun bir arama kategorisi bulunamadı!";
        window.setTimeout(function () {
          aramam.focus();
        }, 0);
      }
    }
  });

</script>

```



```
</body>  
</html>
```

bu dosyamızda kullandığımız media dosyalarını github tan edinebilirsiniz.

Hatırlarsanız settings.py dosyamıza media root ve url eklemiştik, projemize ait olan css, js, jpg, png vs.. gibi dosyaları media klasörümüzde tutacaktık.

```
MEDIA_ROOT          = BASE_DIR + '/media/'  
MEDIA_URL            = '/media/'
```

media linkini kullanabilmemiz için ayrıca url yönlendirmesini de aktif etmemiz gerekiyor.

urls.py dosyamıza aşağıdaki kodlarımızı ekleyelim.

```
from django.conf import settings  
from django.conf.urls.static import static
```

```
urlpatterns = [  
..
```

```
..
```

```
..
```

```
..
```

```
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Terminalde **./manage.py runserver** ile serverımızı başlatalım.

<http://127.0.0.1:8000/>

Alınabilecek hatalar:

SyntaxError: Non-ASCII character '\xc3' in file /home/muslu/django/teknikservistakibi/servisformu/views.py on line 11, but no encoding declared; see <http://python.org/dev/peps/pep-0263/> for details  
views.py dosyasının en başına # -\*- coding: utf-8 -\*- eklemeyi unuttunuz.

Jquery post işlemini kullanarak /kn/ linki ne yazılan takip noyu gönderip dönen bilgiyi sonuclarimuslu divine yazdıracağız.

```
$.post("/kn/", {  
    TakipNo: aramam.value,  
    csrfmiddlewaretoken: "{{ csrf_token }}"  
}, function (data) {  
    sonuclarimuslu.innerHTML = data;  
});
```

Dikkat ederseniz **csrf\_token** kullandık. Bu konuda ileride de hata alabilirsiniz. Önemli bir konu kısaca değinelim ama detaylı bilgi siderseniz aşağıdaki linkte gayet güzel anlatılmış.

<https://www.bilgiguvenligi.gov.tr/web-guvenligi/webdeki-buyuk-tehlike-csrf.html>

Sayfayı okuduktan sonra şöyle bir husus ortaya çıkmış oluyor. Django geliştiricileri bilinen tüm saldırı ve açıklara karşı gerçekten süper bir iş çıkarıyorlar.

Eğer sitenize dışarıdan bir saldırı yapılmak istenirse csrf\_token ile üretilen ve çerezde saklanan bu rastgele değer karşılaştırılır ve sitenize dışarıdan müdahaleye izin vermez.

Csrf ile ilgili ileride daha fazla açıklama yapacağız, şimdiden kafanız karışmasın.

Ana sayfamızı h azırladıktan sonra şimdide views.py dosyamızı açalım ve ana sayfamızdan gelen sorguya cevap için bir yönlendirme yapalım.

```
def formdurumusorgula(request):
```

```
    formno = request.POST.get('TakipNo', None)
```

**### post ile gelen TakipNo'yu formno değişkenine alıyoruz. TakipNo parametresi yoksa None döndür. None yerine istediğiniz Bir şey yazarak yoksa bu olsun diyebilirsiniz.**

Try:

```
    formdurumu = ServisForm.objects.get(FormNo = formno)
```

```
    urunler = Urunler.objects.filter(ServisFormu__id = formdurumu.id)
```

**### gelen formnoya göre veritabanından bilgileri alıyoruz.**

**### bu bilgileri bir string te toplayarak geri göndereceğiz. Bu şekilde sayfa yenilenmeden divin içinde dönen bilgiler gözükecek.**

```
    htmltablo = u"<table style='width:100%;'><tr><th style='width:40%;' align='left'>Ürün</th><th style='width:40%;' align='left'>NOT</th><th style='width:20%;' align='left'>Durum</th></tr>"
```

```
    for k in urunler:
```

```
        htmltablo += "<tr>"
```

```
        htmltablo += "<td align='left'>" + k.Cins.upper() + "(" + k.Marka.upper() + " " + k.Model.upper() + ")</td>"
```

```
        htmltablo += "<td align='left'>" + k.Not + "</td>"
```

```
        htmltablo += "<td align='left'>" + k.Durum.Durumu + "</td>"
```

```
        htmltablo += "</tr>"
```

```
    htmltablo += "</table>"
```

```
    return HttpResponse(htmltablo)
```

**### HttpResponse güzel ama İngilizçe. Buradan ne gönderseniz tarayıcı direk onu basar ekrana.**

except:

```
    return HttpResponse("Form no bulunamadı!")
```

```
    return HttpResponse("Form no bulunamadı!")
```

urls.py dosyamızı açalım ve ana sayfa linki geldiğinde views.py de ki formdurumusorgula fonksiyonunu kullanmak için bir url fonksiyonu yazalım.

```
# -*- coding: utf-8 -*-
from django.conf import settings
from django.conf.urls import url
from django.contrib import admin
from django.conf.urls.static import static
from django.views.generic import TemplateView

from servisformu.views import sayfayiyazdir, formdurumusorgula

urlpatterns = [
    url(r'^$', TemplateView.as_view(template_name="index.html")),
    url(r'^kn/$', formdurumusorgula, name='formdurumusorgula'),
    url(r'^yazdir/([\w\-\+]+)/$', sayfayiyazdir, name='sayfayiyazdir'),
    url(r'^admin/', admin.site.urls),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

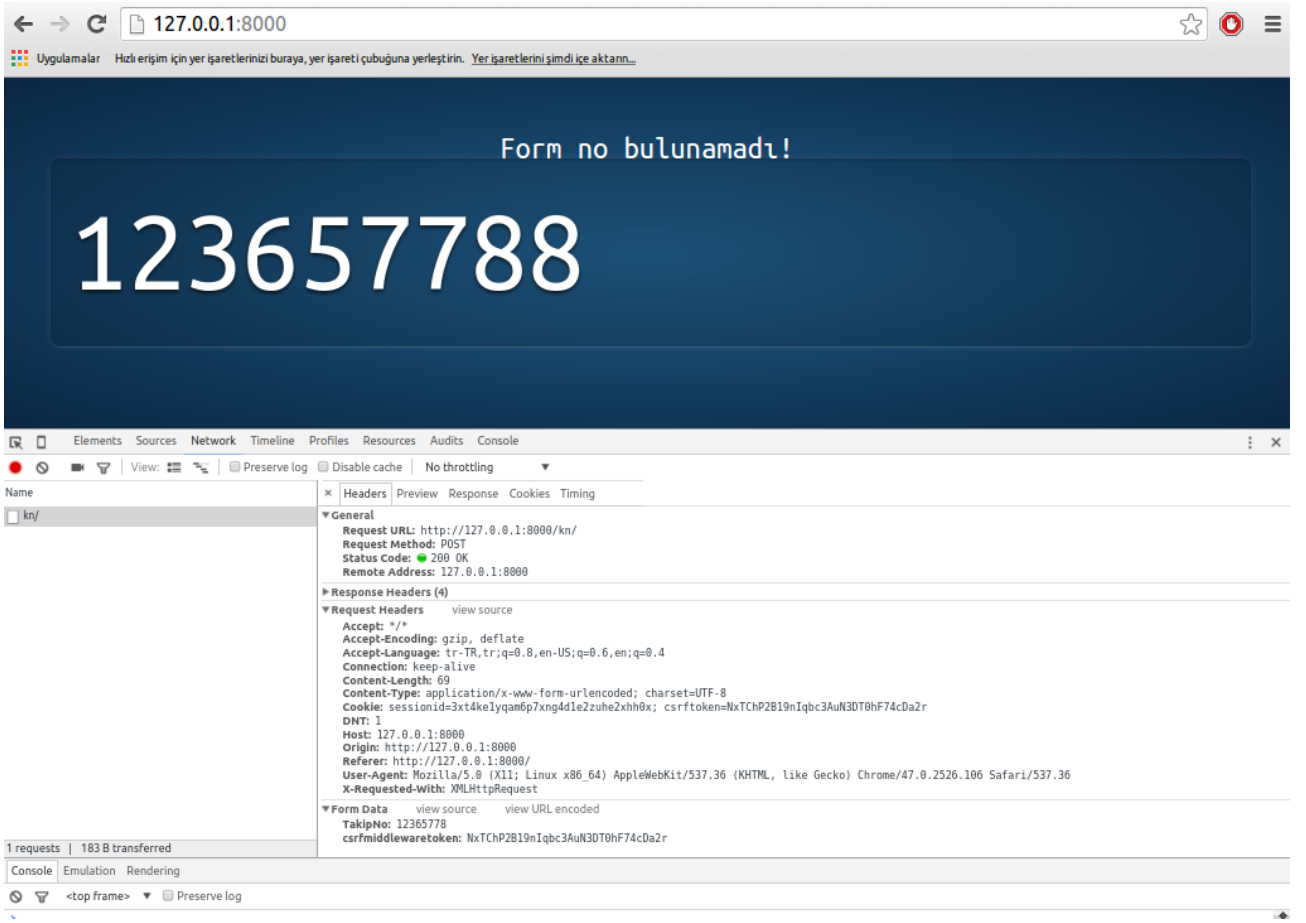
Not:

Önce urls.py yi yada views.py yi hazırlamak size kalmış. Aklınızda gidilecek yolu hazırladıysanız hangisini önce hazırlamak gerektiği önemli değil.

Terminalde:

**./manage.py runserver**

<http://127.0.0.1:8000/>



Ekranın alt ortasında görebileceğiniz gibi Form Data olarak TakipNo ve csrfmiddlewaretoken parametreleri gönderildi.

```

8  def formdurumusorgula(request):
9      formno = request.POST.get('TakipNo', None)
10
11     print formno
12
13     try:
14         formdurumu = ServisForm.objects.get(FormNo = formno)
15         urunler = Urunler.objects.filter(ServisFormu__id=formdurumu)
16         htmltablo = u"<table style='width:100%;'><tr><th st
17
18         for k in urunler:
19             htmltablo += "<tr>"
20             htmltablo += "<td align='left'>" + k.Cins.upper()
21             htmltablo += "<td align='left'>" + k.Not + "</td>"
22             htmltablo += "<td align='left'>" + k.Durum.Durumu
23             htmltablo += "</tr>"
24         htmltablo += "</table>"
25         return HttpResponse(htmltablo)
26     except:

```

Terminal

```

+ Django version 1.9.1, using settings 'teknikservistakibi.settings'
X Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
12365778
[15/Jan/2016 11:04:11] "POST /kn/ HTTP/1.1" 200 20
Performing system checks...

System check identified no issues (0 silenced).

```