

Django

1.9

Middleware

musluyuksektepe@gmail.com

www.yazki.com

www.djangoturkiye.com

Merhabalar;

Django'da sorgu ve yanıt işlemleri katmanlardan oluşur. Basitçe anlatmak gerekirse tarayıcıdan (yada komut ile) bir link çalıştırıldığında `urls.py` de tanımlanan fonksiyon tetiklenir ve bu fonksiyondaki işlemler gerçekleştirildikten sonra bir dosyaya yada http durum yanıtı gönderilir.

Örn:

urls.py

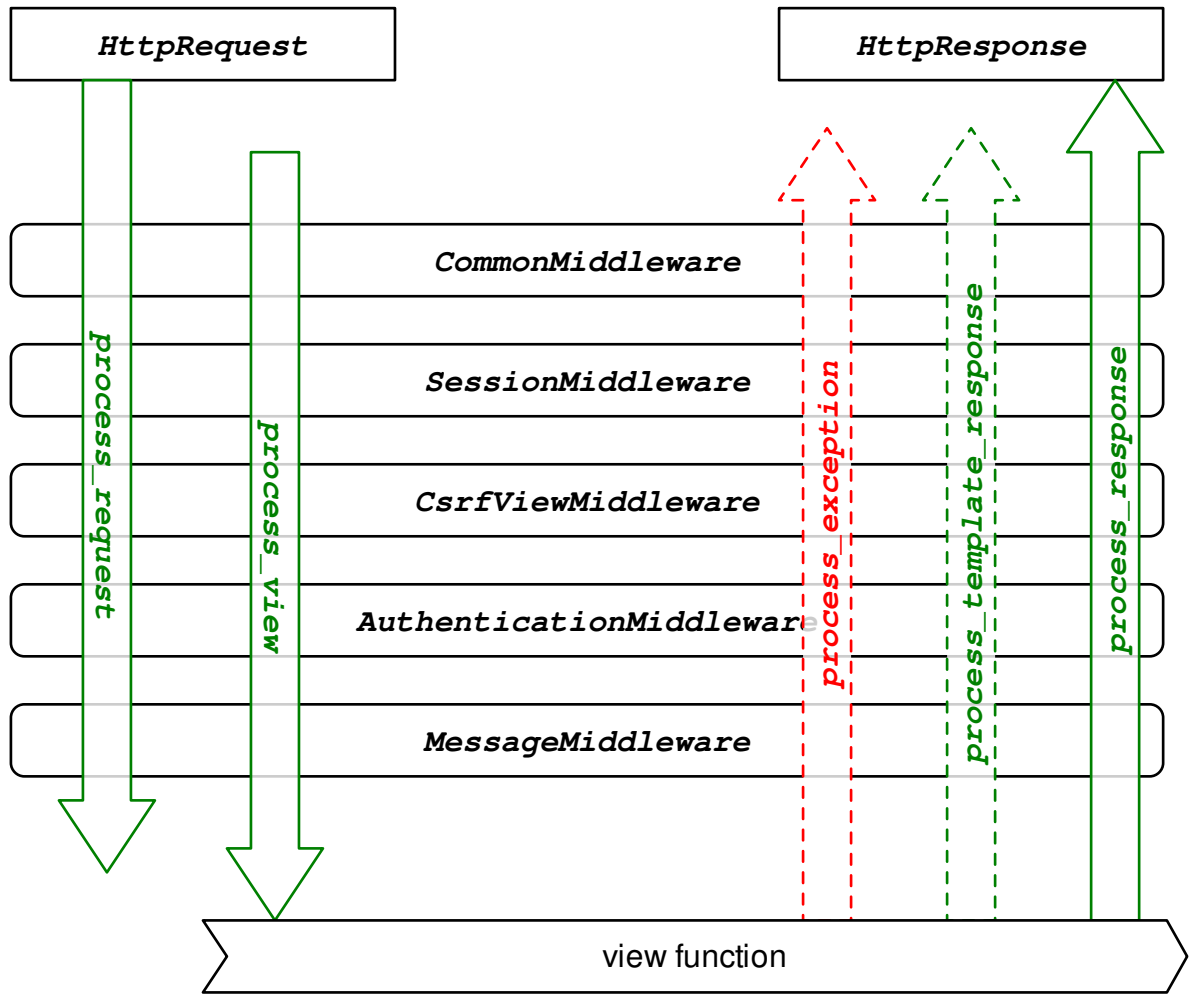
```
urlpatterns = [  
    url(r'^$', ilksayfa),  
]
```

views.py (yada urls.py)

```
from django.shortcuts import render  
def ilksayfa(request):  
    print "anasayfa"  
    return render(request, 'index.html')
```

tarayıcımızda 127.0.0.1:8000 i çalıştırdığımızda önce `urls.py` deki `ilksayfa` fonksiyonu tetiklenir (`url(r'^$', ilksayfa),`) ve `views.py` (yada `urls.py` de olabilir) deki `ilksayfa` fonksiyonu çalıştırılır ve `return` ile bir yanıt gönderilir.

Peki; tüm sayfalarda var olmasını istediğimiz bir işlem varsa bunun için tüm fonksiyonlara bu işlemi yazıp göndermek mi gerekir? Hayır. Django daki middleware (katman) mantığı bunun için var.



Görmüş olduğunuz gibi örnekte belirttiğim bir sorgu yapıldığında katmanlar arasında işlemler tamamlanır ve fonksiyona ulaşır daha sonra da istenilen cevap geri gönderilir.

Pekiştirmek için biz bir örnek katman oluşturup kendi yanıtlarımızı verelim. Github dan projenin tamamını indirebilirsiniz.

Herhangi bir uygulamanızın altında `ozel_islemler.py` adında bir dosya oluşturup alttaki kodları yazın.

Standartta dosya adı `middleware.py` olması tavsiye edilir

ama kalıplaşma hissetmeyin diye farklı bir ad örneği verdim.

/home/muslu/django/uyegrup/uyeler/ozel_islemler.py

ozel_islemler.py

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

```
from django.core.exceptions import PermissionDenied
```

```
class IPKontrolMiddleware(object):
```

```
    def process_request(self, request):
```

```
        iplistesi = ['192.168.1.1', '127.0.0.1', '0.0.2.35']
```

```
        ip = request.META.get('REMOTE_ADDR')
```

```
        print "process_request = sorgu, ", ip
```

```
        if ip not in iplistesi:
```

```
            raise PermissionDenied
```

```
        return None
```

```
    def process_response(self, request, response):
```

```
        print "process_response = yanıt, ", request.user
```

```
        return response
```

settings.py

```
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',  
    'uyeler.ozel_islemler.IPKontrolMiddleware'  
]
```

Not: middleware tanımlamalarında sıralama önemlidir. Kullanacağınız bilgiler önceden tanımlanmış, işlenmiş, hazırlanmış vss olması gerekir.

Şimdi bu katmanda neler yaptığımıza bakalım.

Katmanımızın adı IPKontrolMiddleware, iki adet alt fonksiyon kullandık. Detaylarını django proje sayfasında bulabilirsiniz. Cevap ve yanıt. Ayrıca bir de ana sayfa çağrıldığındaki işlem sırasını göreceğiz.

Sorgu tetiklendiğinde yapmayı istediklerimiz;

```
def process_request(self, request):  
    ### sorgu başladığında request argümanına  
    ihtiyacımız var  
  
    iplistesi = ['192.168.1.1', '127.0.0.1', '0.0.2.35']  
    ### İzin vereceğimiz ip adresleri  
  
    ip = request.META.get('REMOTE_ADDR')  
    ### ip adresimiz  
  
    print "process_request = sorgu, ", ip  
    ### işlem sırasını ekranda görmek için fonksiyon adı  
    ve ip adresimiz  
  
    if ip not in iplistesi:  
        ### ip adresimiz izin verilen listesinde yok ise  
        raise PermissionDenied  
        ### ekrana 403 Forbidden yazdıracağız. Farklı  
        şeyler deneyerek iyice kavramaya çalışın.  
  
    return None  
### ip adresimiz listede varsa herhangi bir şey yapma.  
Burada None göndermezseniz hata alırsınız.
```

```
def process_response(self, request, response):  
    ### sorgu geldikten sonra views.py de ki sorgumuz çalıştı  
    ve artık yanıtı gönderirken istediğimiz şeyleri yapabiliriz.  
    Burada request ve response argümanlarını kullandık
```

```
    print "process_response = yanıt, ", request.user  
    ### ekrana yanıtın yapıldığını bastıracağız.
```

```
    return response
```

```
    ### response u geri gönderiyoruz.
```

Tüm adımları (settings.py, özel_islemler.py, views.py) tamamladıysanız şimdi bir sorgu tetikleyip sırayla neler olduğuna bakalım.

May 09, 2016 - 14:27:50

Django version 1.9.4, using settings 'uyegrup.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CONTROL-C.

process_request = sorgu, 127.0.0.1

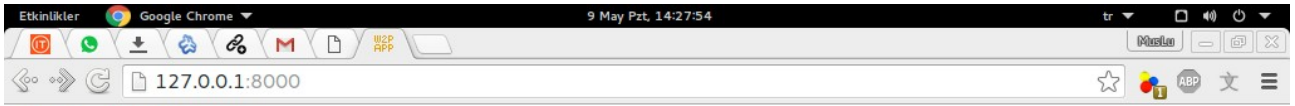
Forbidden (Permission denied): /

process_response = yanıt, muslu@muslu.org

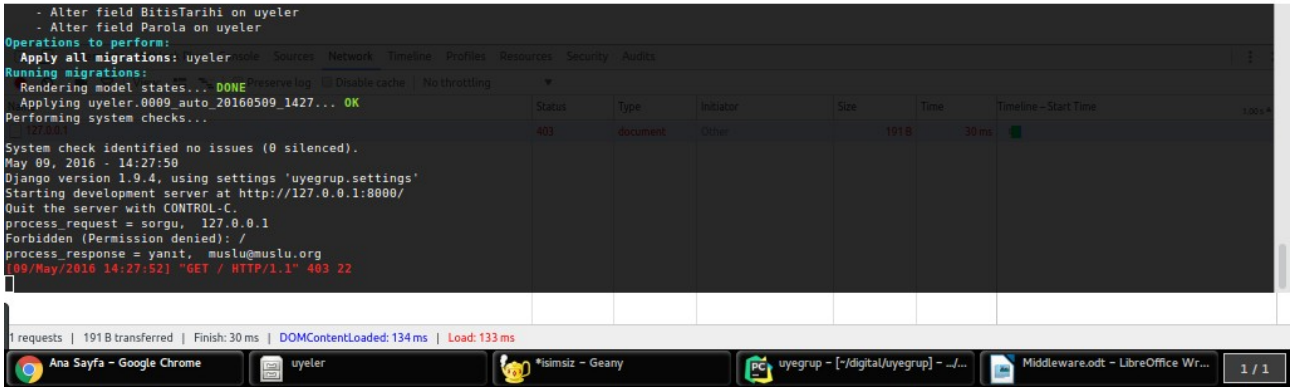
[09/May/2016 14:27:52] "GET / HTTP/1.1" 403 22

Sırayla önce **ozel_islemler.py** dosyamızda tanımladığımız **IPKontrolMiddleware** fonksiyonumuz devreye girdi ve **process_request** sorgumuz çalışıp ekrana **process_request = sorgu, 127.0.0.1** yazdı ve daha ilksayfa fonksiyonumuzu tetikletmeden ip adresimizin listede olmadığını görünce hemen forbidden uyarısı verdi ve yanıtı gönderdi.

1. özel_islemler.py
2. IPKontrolMiddleware
3. process_request
4. process_response



403 Forbidden



Şimdi de ip adresimize izin vererek deneyelim.

ozel_islemler.py de ki iplistesi adlı listemizi değiştirip
127.0.0.1 ekleyelim.

Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

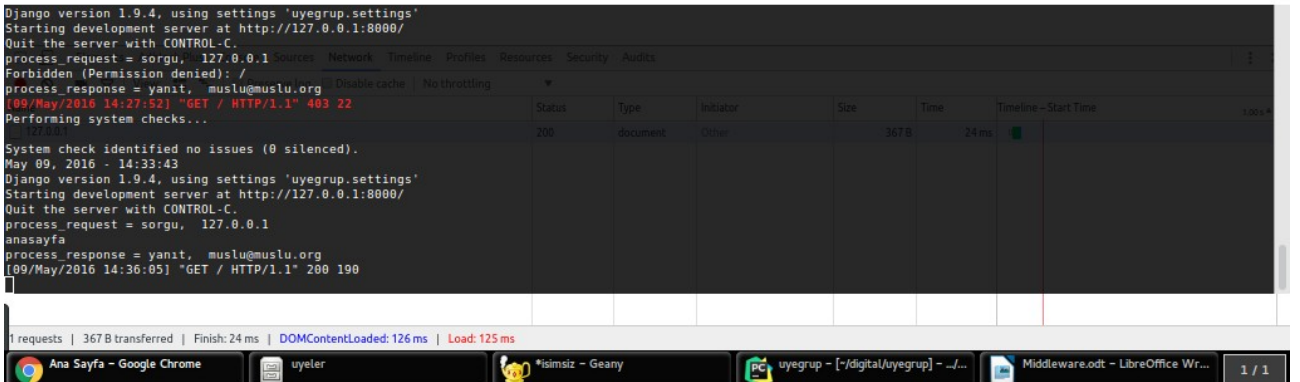
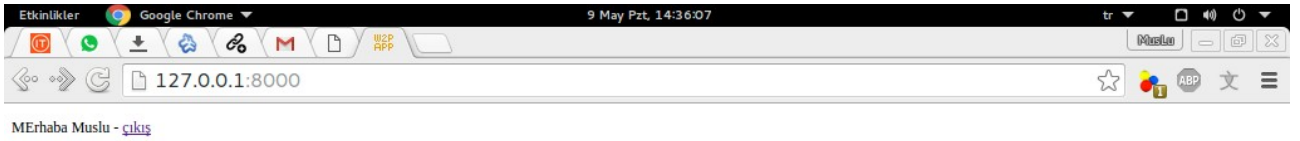
process_request = sorgu, 127.0.0.1
anasayfa

process_response = yanıt, muslu@muslu.org

[09/May/2016 14:36:05] "GET / HTTP/1.1" 200 190

Görüldüğü gibi yine katmanımız çalıştı ve sorgu geldiğini göstermek için ekrana ***process_request = sorgu, 127.0.0.1*** yazdı daha sonra ip adresimiz listede olduğu için urls.py den ilksayfa nın views.py deki fonksiyonuna gitti ve ekrana ***anasayfa*** yazıp tekrar katmanımızdaki yanıt fonksiyonunu tetikleyip ***process_response = yanıt, muslu@muslu.org*** yazdırdık.

1. özel_islemler.py
2. IPKontrolMiddleware
3. process_request
4. urls.py
5. views.py (ilksayfa fonksiyonunu urls.py de de yazabilirsiniz.)
6. process_response



Farklı bir link tetikletip tüm sayfalarda çalışıp çalışmadığına bakalım.

Views.py dosyamızı açıp üye girişi ve çıkışı için 2 yeni fonksiyon ekleyelim.

views.py

```
@login_required(login_url='/giris/')
def ilksayfa(request):
    print "anasayfa"
    return render(request, 'index.html')
```

```
def cikis(request):
    logout(request)
    print "cikis, anasayfaya yönlendir"
    return redirect("/")
```

```
@csrf_protect
def giris(request):
    logout(request)
    mail = parola = ""
    if request.POST:
        mail = request.POST['mail']
        parola = request.POST['parola']
```

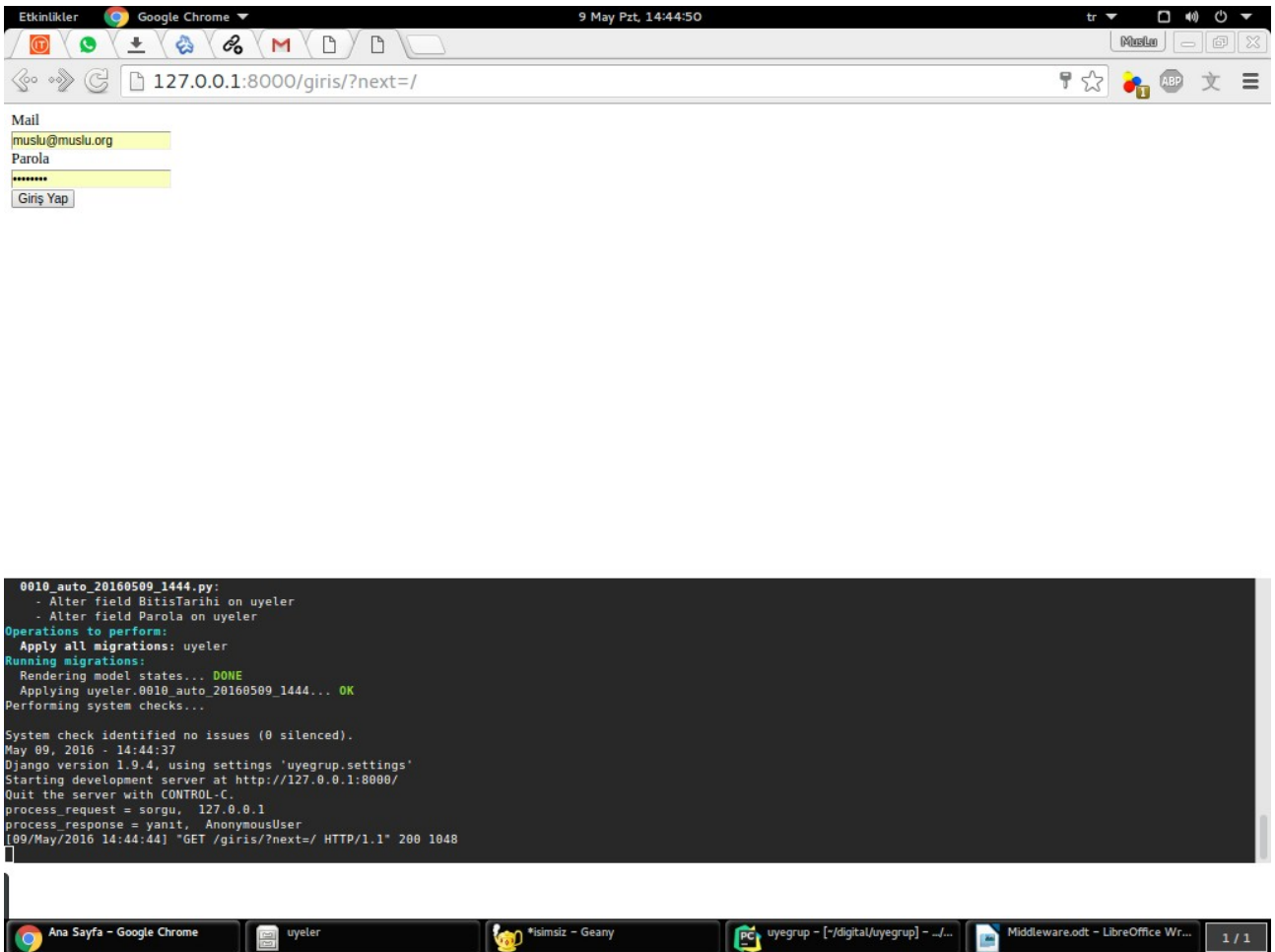
```
    print "giris, anasayfaya yönlendir, ", mail, parola
```

```
    user = authenticate(username=mail, password=parola)
    if user is not None:
        if user.is_active:
            login(request, user)
            return HttpResponseRedirect('/')
    return render(request, 'giris.html')
```

urls.py

```
urlpatterns = [  
    url(r'^$', ilksayfa),  
    url(r'^giris/$', giris),  
    url(r'^cikis/$', cikis),  
    url(r'^admin/', admin.site.urls),  
]
```

Şimdi tarayıcımızda bu iki linki de deneyelim.



Starting development server at <http://127.0.0.1:8000/>

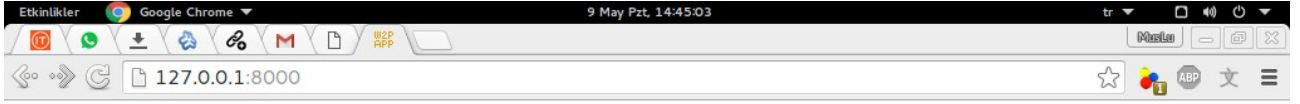
Quit the server with CONTROL-C.

process_request = sorgu, 127.0.0.1

process_response = yanıt, AnonymousUser

[09/May/2016 14:44:44] "GET /giris/?next=/ HTTP/1.1" 200 1048

Ana sayfayı çağırdığımızda giriş yapmamız gerektiği için (`@login_required(login_url='/giris/')`) giris sayfasına yönlendirdi. Bu arada katmanımızdaki sorgu ve yanıt normal şekilde çalıştı ve giriş yapılmadığı için `request.user` cevabı olarak ***AnonymousUser*** yazdı.



MERhaba Muslu - çıkış

```
Performing system checks...
System check identified no issues (0 silenced).
May 09, 2016 - 14:44:37
Django version 1.9.4, using settings 'uyegrup.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
process_request = sorgu, 127.0.0.1
process_response = yanıt, AnonymousUser
[09/May/2016 14:44:44] "GET /giris/?next=/ HTTP/1.1" 200 1048
process_request = sorgu, 127.0.0.1
giris, anasayfaya yönlendir, muslu@muslu.org TX1J8PRO
process_response = yanıt, muslu@muslu.org
[09/May/2016 14:45:01] "POST /giris/ HTTP/1.1" 302 0
process_request = sorgu, 127.0.0.1
anasayfa
process_response = yanıt, muslu@muslu.org
[09/May/2016 14:45:01] "GET / HTTP/1.1" 200 190
```

[09/May/2016 14:44:44] "GET /giris/?next=/ HTTP/1.1" 200 1048

process_request = sorgu, 127.0.0.1

giris, anasayfaya yönlendir, muslu@muslu.org TX1J8PRO

process_response = yanıt, muslu@muslu.org

[09/May/2016 14:45:01] "POST /giris/ HTTP/1.1" 302 0

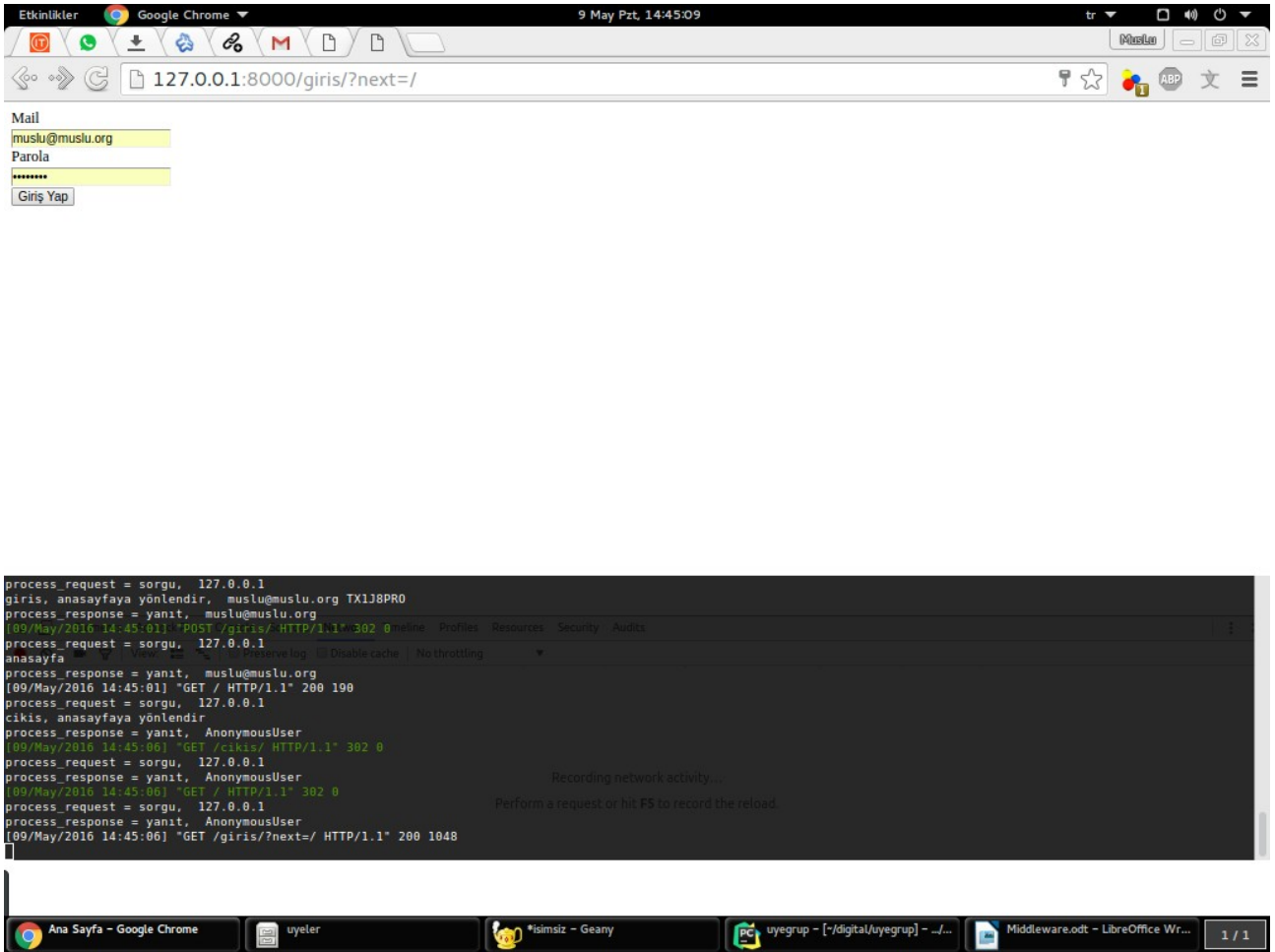
process_request = sorgu, 127.0.0.1

anasayfa

process_response = yanıt, muslu@muslu.org

[09/May/2016 14:45:01] "GET / HTTP/1.1" 200 190

giriş butonuna basıldığında katmanımızdaki sorgu çalıştı, giriş fonksiyonumuz tetiklendi ekrana bilgilerimizi yazdı ve katmanımızdaki cevap çalıştı. Ana sayfaya yönlendirildik ve katmanımızdaki sorgu, anasayfa fonksiyonumuzdaki print ve katmandaki yanıt çalıştırdı.



```
[09/May/2016 14:45:01] "GET / HTTP/1.1" 200 190  
process_request = sorgu, 127.0.0.1  
cikis, anasayfaya yönlendir  
process_response = yanıt, AnonymousUser  
[09/May/2016 14:45:06] "GET /cikis/ HTTP/1.1" 302 0  
process_request = sorgu, 127.0.0.1  
process_response = yanıt, AnonymousUser  
[09/May/2016 14:45:06] "GET / HTTP/1.1" 302 0  
process_request = sorgu, 127.0.0.1  
process_response = yanıt, AnonymousUser  
[09/May/2016 14:45:06] "GET /giris/?next=/ HTTP/1.1" 200 1048
```

Buradaki yorumu kendiniz yapmaya çalışın.

Böylelikle Django'da ki arkaplanda neler oluyor sorusuna cevap vermiş olduk. Daha farklı fonksiyonlar kullanarak denemeler yapabilirsiniz. Örnek olarak yanıt yada sorgu fonksiyonuna `request.path` ekleyerek hangi sayfadan gelip yönlendiğini görebilirsiniz.