

# Url işlemleri

14 Ağustos 2016 Pazar

14:28



url(r'^\$', home)	Ana sayfa
r' '	Tırnak içine yazılacak kısım regextir diye adlandırıyor
^	Regex in başlama kısmı
\$	Regex in bitme kısmını gösteriyor



Detail sayfasını oluşturmak için **view.py** kısmına şu şekilde kod yazıyoruz :

- def blog\_detail(request, id):  
    post = Post.objects.get(id = id)  
    return render(request, "detail.html", {  
        "post" : post  
    })



**Anasayfa'dan url yönlendirmesi** için html içinde href kodu olarak bunu giriyoruz :

- <a href="**posts/{{ post.id }}**"> {{ post.title }} </a>



Bunun yerine şöyle de girebiliriz :

- href=" {% url "post\_detail" post.id %}"



Başka bir yöntem :

- Url.py kısmında :
  - url(r'^places/(?P<id>[-\w]+)/\$', place\_detail, name="place\_detail")
    - Burada url'yi adlandırıyoruz
- Model kısmında :
  - class Place (models.Model):
    - @models.permalink
    - def get\_absolute\_url(self):
      - return ("place\_detail", (self.id,))
      - Burada adlandırdığımız url'yi çekip, id ekleyerek yeni bir link oluşturuyoruz.
- Html kısmında :
  - <a href="{**place.get\_absolute\_url**}">{{ place.name }}</a>
    - Oluşturduğumuz linki fonksiyon olarak burada çekiyoruz.



Url.py dosyası içinde, gelen url olarak böyle bir kod girmemiz gerekiyor (**GET - ID çekme**) :

url(r'^posts/(?P<id>[0-9]+)/\$', blog_detail)	Sadece sayısal değerler alabilir.
url(r'^posts/(?P<id>[-\w]+)/\$', blog_detail)	Bu parametre herhangi bir değer alabilir.



Bu yukarıdaki gösterim id aralık dışındaysa hata verir. Bu hatadan kurtulmak ve **404 hatası** vermesini sağlamak için şu yöntemi kullanabiliriz :

- View.py içinde :
  - from django.shortcuts import render, **get\_object\_or\_404**
  - def blog\_detail(request, id):

```
    post = get_object_or_404(Post, id = id)
    return render(request, "detail.html", {
        "post" : post
    })
```