برای این url

url(**r'^users/(?P<user\_id>[0-9]+)/$'**, views.detail\_user, name=**'users'**),

نوشته بودم :

<**a href="/persons/users/**{{**owner**.**id**}}**/"**>{{**owner**.**username**}}</**a**>

**اما تمیزتر این است که اینگونه بنویسم:**

**<a href="{% url 'persons:users' user\_id=owner.pk %}">{{owner.username}}</a>**

**یعنی منظورم این بود که که کلید پارامتر را هم به راحتی میتوان فرستاد**

یک url برای گرفتن کلید اصلی:

url(r'(?P<pk>\d+)/$',views.course\_detail),

یک url برای گرفتن دو کلید اصلی:

url(r'(?P<course\_pk>\d+)/(?P<step\_pk>\d+)/$',views.step\_detail,

name='step'),

حالا چطور به این url دو پارامتری ارجاع بدهیم:

<a href="{% url 'courses:step' course\_pk=step.course.pk step\_pk=step.pk %}">{{ step.title }}</a>

البته راه دیگری هم به جای استفاده از تمام این وجود دارد و آن این است که متد get\_absolute\_url را معرفی کنیم.

به این url نگاه کن :

url(r'(?P<course\_pk>\d+)/t(?P<step\_pk>\d+)/$',views.text\_detail,

name='text'),

حالا به این متد که در کنار مدل معرفی کرده ام نگاه کن:

def get\_absolute\_url(self):

return reverse('courses:text',kwargs={

'course\_pk':self.course\_id,

'step\_pk':self.id

})

تو حتما می دانی که courses از کجا امده . ان چه در این کار جالب است این است که نگاه کن که چطور به پارامتر ها به سادگی مقدار داده. اصلا شاید دلم بخواهد دیگر من هم همه را این طوری بنویسم از بس خوب و تمیز است. به ان حرف t که در url امده نگاه کن که چطور هیچکاره است و فقط خیلی قشنگ فقط به کاربر نشان داده می شود و برای ما هنگام کدنویسی گویی اصلا وجود ندارد. و به راحتی می توانی ان ها را تغییر دهی بدون اینکه نیاز باشد چیز دیگری را تغییر دهی.

حالا چگونه از این در صفحات template استفاده کنیم:

{% for step in steps %}

<a href="{{step.get\_absolute\_url}}">{{ step.title }}</a>

{% endfor %}

چقدر خوب است این طرز کد نویسی ادم عشق میکند. به نظر من این روش خوبی است که هرکلاسی خودش مشخص کند اگر روی شی ای از وی کلیک شد باید به کجا هدایت شویم.

الان یک چیز حیرت اور فهمیدم این مرد ریشو اعجوبه ایست برای خودش تازه دارم می فهمم که چه کرده. این مرد می اید دو نوع کوئری را با هم در یک متغیر می ریزد و ان را به ویو می فرستد انگاه وقتی روی ان متغیر for می نوسید چون به روش فوق عمل کرده هرگدام می دانند کجا باید بروند

در view.py چگونه از name هایی که به url ها داده ایم استفاده کنیم:

from django.core.urlresolvers import reverse

from django.http import HttpResponseRedirect

return HttpResponseRedirect(reverse('suggestion'))

که suggestion از کد زیر امده:

url(r'^suggest/$', views.suggestion\_view,name='suggestion'),

فعلا نمی دانم اگر کار به پارامتر کشید چه باید کرد. فک کنم از ان get\_absolute باید استفاده کرد.

Examples:

<a href="{{comment.get\_absolute\_url}}delete" class='btn btn-default'>delete</a>

جاستین وقتی که دارد اموزش rest می دهد می گوید که ما همیشه pk را با این url می گیریم :

url(r'^(?P<pk>\d+)/$', PostDetailAPIView.as\_view(), name='detail'),

با توجه به اینکه مرد ریشو هم همین کار را کرده پس جای تردید برای ما نمی ماند که ما هم pk ها را باید با این url بگیریم.

رفع یک خطای اعصاب خورد کن:

می خواستم پس از ثبت نام وارد صحفه ی لاگین شویم این خط

return HttpResponseRedirect('login')

مارو می برد به

<http://127.0.0.1:8000/accounts/login/>

که وجود نداشت و من امدم این خط را نوشتم:

return redirect('login')

که دیگر مارو می برد به

<http://127.0.0.1:8000/login/>

و درست هم بود. منظور این بود که دو متد فوق با هم فرق هایی دارند.

ایجاد لینک:

راه های مختلفی برای ایجاد لینگ هست

**return** HttpResponseRedirect(reverse(**'cart'**))

**ما را به cart می بردد.**

**return** HttpResponseRedirect(**'/'**)

این مارا به صفحه ای اصلی میبرد.

یک روش خوب برای مقدار دادن به تگ a بلدم :

<a href="{% url 'products' %}">products</a>

Products نامی است که به یو ار ال داده ایم در urls.py

حالا اگر دو قسمتی بود مثلا بخواهیم همچین چیزی را پیاده کند persons/show\_homes و یک نام برای persons داشته باشیم و نام دیگری برای show\_homes این گونه میشود:

<a href="{% url 'persons:show\_homes' %}">Project name</a>

شاید بخواهی بدانی که get\_absolute\_url استفاده کرده ام

class Member(models.Model):

user = models.OneToOneField(User, on\_delete=models.CASCADE)

phone\_number = models.CharField(max\_length=10)

national\_code = models.CharField(max\_length=10)

def get\_absolute\_url(self):

return "/persons/users/%s/" %self.user.pk

وقتی این را نوشته بودم دیگر می توانستم در متد redirect شی ای از کلاس member بگذارم دیگر خودش به صورت پیشفرض متد get\_absolute\_url را صدا می زد. حالا جالب تر اینکه من برای اینکه به صفحه ی کاربر بروم id کاربر را می خواستم پس از رابطه ی یک به یک بین member و user استفاده کردم و این طوری کارم را راه انداختم

return redirect(\_member)

که \_member شی ای از نوع کلاس member است. که طبق متد get\_absolute\_url که برایش تعریف کردم شماره کاربر مربوطه را برمی گرداند.