در فولدر py36\_dj110 یک فولدر به نام forms وجود دارد که این فولدر قرار است برای من اساس کار با فرم های جنگو باشد و من کدهای اماده در انجا داشته باشم . فرم ها در لینک زیر نشان داده میشوند:

<http://127.0.0.1:8000/suggest/>

برای فرم چه کلماتی داریم:

subject = forms.CharField(max\_length=100, help\_text='100 characters max.')

sender = forms.EmailField(help\_text='A valid email address, please.')

cc\_myself = forms.BooleanField(required=False)

import datetime

day = forms.DateField(initial=datetime.date.today)

url = forms.URLField(initial='http://')

name = forms.CharField(initial='Your name')

یادمه یک روز میخواستم به صورت دستی خودم فیلدی برای text بسازم اما ازم برنمی امد :

text = forms.CharField(widget=forms.Textarea)

برای این که csrp برایمان کار کند باید کوکی ها ست شده باشند.

------------------------------------------------------------------------------------

برای فرم ها چطور اعتبار سنجی هایی داریم؟ انواع انها را مرد ریشو اموزش می دهد.

* می توانیم clean\_name بنویسیم
* می توانیم از validator های جنگو استفاده کنیم
* میتوانیم خودمان validator بنویسیم
* علاوه بر نوشتن clean روی یک فیلد خاص فرم می توان روی کل فرم هم یک متد clean نوشت. در این صورت لازم نیست که حتما فیلدی خاص را return کنیم.
* -------------------

اینکه در متد هایی که برای اعتبارسنجی می نویسیم چگونه به مقادیری که کاربران در فرم وارد کرده اند دسترسی داشته باشیم:

برای اینکار دو راه در اموزش مرد ریشو دیده ام:

honeypot = self.cleaned\_data['honeypot']

email = cleaned\_data.get('email')

و یک راه هم که جاستین استفاده می کند:

email = self.cleaned\_data.get("email")

البته شباهت انها مبرهن است.

چرا پیام هایی که برای خطاهای اعتبارسنجی نوشته بودم نمایش داده نمی شدند:

علت این بود که دستور زیر را در views نوشته بودم:

return HttpResponseRedirect(reverse('suggestion'))

این دستور صفحه ی جدیدی باز می کند پس معلوم است که error ها دیگر نمایش داده نمی شوند. این دستور باید در شرط valid بودن فرم گذاشته شود. خطاهای مربوط به فرم با message ها فرق دارند ان ها حتی اگر صفحه هم غوض شود باز نمایش داده می شوند زیرا پیام نمایش انهارا در template کلی گذاشته ایم.

چگونه فیلدی را تاریخی کنیم:

publish = forms.DateField(widget=forms.SelectDateWidget)

print(dir(form))

print(form.errors)

یک بار دیگر یادآوری می کنم:

**if** request.method == **'POST'**:

form = forms.ResearchForm(request.POST,request.FILES)

**print** (form.errors)

**if** form.is\_valid():

با کدهای فوق می‌توان دید که چرا فرم

valid

نیست یعنی اینکه چرا از متد

is\_valid

عبور نمی کند

استراتژی های کار با فرم ها:

برای اینکه در یک فرم با کلاس مدل کار کنیم لازم نیس حتما model form داشته باشیم می توان فرم را standard ساخت اما کلاس مدل را هم import کرد و در متدهای clean و اعتبار سخنجی روی مدل ها کوئری ززد و یا متدهایشان را صدا زد مثل اموزش advanced blog که در ان جاستین کلاس user را ایمپورت می کند .

قالب ساده ی من برای رندر کردن فرم این است:

<form method="post"enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

<input type="submit" value="Submit" />

</form>

فرمت کار با فرم ها این چنین است:

def create:

form()

if request.POST:

form(request.POST)

if form.is\_valid():

save()

redirect()

return(request,form)

def update:

instance= Query

form(instance)

if request.POST:

form = form(request.POST, instance= instance)

if form.is\_valid():

form.save()

redirect()

return (request, form)

در مورد فرم ها یک الگوی دیگر پیدا کرده ام که شاید ان هم خوب باشد

<fieldset>

<legend>Package <small>how your account will be billed</small></legend>

<div class="control-group {% if form.package.errors %}error{% endif %}">

<label class="control-label">{{ form.package.label }}</label>

<div class="controls">

<ul>

{% for choice in form.package.field.choices %}

<li>

<input type="radio"

name="{{ form.package.name }}"

value="{{ choice.0 }}"

{% if form.package.value"|safe == choice.0|safe %}

checked="checked"

{% endif %}

>

<strong>{{ choice.2.name }}</strong>

<span class="price">{{ choice.2.price }}</span>

</li>

{% endfor %}

</ul>

{% if form.package.errors %}

<span class="help-block">{{ form.package.errors }}</span>

{% endif %}

</div>

</div>

</fieldset>