

# آموزش جنگو مهران تعریف

۲۹ شهریور ۱۴۰۱

## فهرست مطالب

۳	۱	قسمت ۵۱ نوشتن mixin های شخصی برای مدیریت بهتر کد
۳	۱.۱	Bug Fix
۳	۲.۱	فقط سوپر یوزر بتواند فیلد های نویسنده و وضعیت یک مقاله را مشخص کند
۳	۳.۱	فیلد نویسنده به صورت پیشفرض برای نویسنده مقدار داشته باشد
۴	۲	قسمت ۵۲ ساخت ویو آپدیت
۴	۱.۲	پیاده سازی
۴	۲.۲	رفع باگ
۴	۱.۲.۲	کد نویسی
۴	۳.۲	رفع باگ ۲
۵	۱.۳.۲	کدنویسی
۵	۳	قسمت ۵۳ ساخت ویو حذف مقالات و میکسین دسترسی مدیران
۵	۴	قسمت ۵۴ ساخت ویو لاگ اوت
۵	۱.۴	رفع باگ
۵	۵	قسمت ۵۵ اکتیو منو و حذف مقاله از بخش ویرایش
۵	۱.۵	حذف مقاله در قسمت ویرایش
۵	۲.۵	چجوری template tags های خودمون بسازیم
۶	۳.۵	چجوری منوی اکتیو رو فعال کنیم داخل ساید بار
۶	۶	قسمت ۵۶ ساخت پیشنمایش مقاله های پیشنهادی در پنل مدیریت

## ۱ قسمت ۵۱ نوشتن mixin های شخصی برای مدیریت بهتر کد

### ۱.۱ Bug Fix

لینک به اسم نویسنده داخل صفحه اصلی زیر مقاله ها درست کار نمیکنه چون داخل ویو کلاس یوزر خود جنگو لود میکنه ولی ما از کلاس یوزر شخصی استفاده میکنیم (داخل تمپلیت ها هم به جای اشتباهی لینک شده بود و خود تمپلیت هم همه مقاله ها را نشان میداد هم منتشر شده هم بایگانی) و استریپ کردن تگ ها داخل صفحه اصلی و صفحه مقالات نویسنده

### ۲.۱ فقط سوپر یوزر بتواند فیلد های نویسنده و وضعیت یک مقاله را مشخص کند

برای پیاده سازی این قابلیت از میکسین ها استفاده میکنیم هم کد تکراری کم میشه هم نگهداری اسان تر میشه. یک فایل mixins.py داخل پوشه اپ میسازیم و یک کلاس داخل درست کنیم داخل کلاس متد dispatch را override میکنیم dispatch متدی هست که ریکوست رو میگیره و یک ریسپانس برمیگردونه می خوایم بگیم اگر یوزر ریکوست سوپر یوزر بود فیلدهای عادی، وضعیت و نویسنده را به فرم اضافه کن در غیر این صورت اگر نویسنده بود فقط فیلد های عادی رو اضافه کن و در اخر اگر هیچ یک نبود صفحه را نمایش نده (ارور بده بهش) میکسین را در فایل views.py ایمپورت میکنیم و کلاس مربوط (ساخت مقاله) ازش ارث بری میکنه

```
from django.http import Http404

class FormFieldsMixin:

    def dispatch(self, request, *args, **kwargs):
        if request.user.is_superuser:
            self.fields = ['title', 'slug', 'category',
                           'description', 'thumbnail', 'publish', 'status', 'author']

        elif request.user.is_author:
            self.fields = ['title', 'slug', 'category',
                           'description', 'thumbnail', 'publish',]

        else:
            raise Http404('امایش غیرمجاز')
        return super().dispatch(request, *args, **kwargs)
```

داخل فایل تمپلیت هم فیلد های نویسنده و وضعیت که بهشون استایل دادیم داخل یک شرط قرار بگیره که تنها وقتی که کاربر سوپر یوزر هست نمایش داده بشه

### ۳.۱ فیلد نویسنده به صورت پیشفرض برای نویسنده مقدار داشته باشه

چون نویسنده نمیتونه فیلد های نویسنده و وضعیت یک مقاله را تعیین کند میخوایم زمانی که یک مقاله اضافه کرد اون مقاله به صورت دیفالت نویسندهش بشه همون نویسنده ای که مقاله را نوشه و وضعیت هم draft باشه به صورت خودکار برای این کار از از یک میکسین و یک متد form valid استفاده میکنیم و مشخص میکنیم اگه کاربر نویسنده عادی بود به صورت خود کار نویسنده را برابر نویسنده مقاله قرار بده و از وضعیت draft استفاده کن بعد از میکسین داخل ویو مربوطه استفاده میکنیم برای اینکه بتونیم مقدار فیلد هارو داخل فرم تغییر بدیم از commit=False استفاده میکنیم.

```
class FormValidMixin:
    def form_valid(self, form):
        if self.request.user.is_superuser:
            form.save()
        else:
            self.obj = form.save(commit=False)
            self.obj.author = self.request.user
            self.obj.status = 'D'
            form.save()

    return super().form_valid(form)
```

## ۲ قسمت ۵۲ ساخت ویو آپدیت

### ۱.۲ پیاده سازی

مثل همون ویو کریت هست ولی بجای create از update ارث بری میکنه (امپورت بشه) به یو ار ال ها اضافه بشه داخل یو ار ال از primary key استفاده کنیم و داخل تمپلیت ها هم از لینک روی اسم مقاله استفاده کنیم (چجوری ایدی پاس بدیم به یو ار ال داخل تمپلیت)

### ۲.۲ رفع باگ

کاربران میتوانند با استفاده از ایدی مقاله داخل یو ار ال مقاله های پابلیشو درفت کنند با تغییر اون یا مقاله های کس دیگه ای رو تغییر بدن باید یه دسترسی داشته باشیم که هر کاربر فقط بتونه مقاله های خودشو تغییر بده اگر سوپر یوزر نبود ولی اگر سوپر یوزر بود مقاله های بقیه هم بتونه تغییر بده

### ۱.۲.۲ کد نویسی

۱. بدون میکسین جدید میسازیم و از متد dispatch استفاده میکنیم

۲. با استفاده از متد get\_object\_or\_404 مقاله را از دیتابیس میخوانیم

۳. نویسنده مقاله را با یوزر ریکوست چک میکنیم

۴. شروط را اعمال میکنیم

داخل ویو از این میکسین بجا LoginRequired استفاده میکنیم چون مطمئن هستیم دیگه کاربر لاگین هست

### ۳.۲ رفع باگ ۲

کاربران عادی میتوانند با تغییر دادن مقاله ی منتشر شده وضعیت مقاله را به درفت عوض کند میخوانیم فقط بتوانند مقاله های درفت را تغییر بدهند

## ۱.۳.۲ کدنویسی

داخل متد `dispathch` در میکسین شرطمون رو اینجوری مینویسیم که کاربر عادی باشد و مقاله درفت باشد یا کاربر سوپر یوزر باشد و داخل تمپلیت هم فقط نام مقاله هایی لینک شده باشند که وضعیت درفت دارند یا کاربر سوپر یوزر باشد.

## ۳ قسمت ۵۳ ساخت ویو حذف مقالات و میکسین دسترسی مدیران

جنگو دیلیت ویو داکيومنتس از خود مثال داکيومنت استفاده میکنیم بخش ریورس لیزی مارو ببره به صفحه مقالات داخل پنل ادمین، یو ار ال اضافه کنیم، داخل صفحه مقالات یه بدج اضافه بشه برای حذف ولی فقط برای سوپر یوزر، یدونه میکسین هم برای دیلیت اکسس (سوپر یوزر اکسس) میسازیم

## ۴ قسمت ۵۴ ساخت ویو لاگ اوت

از ادمین اصلی نوبار بالا کپی میکنیم دانلود ایکون مجانی لاگ اوت از `awesome` تمپلیت به صورت پیشفرض از `awesome` استفاده میکنه (مجانی هاش) ان کامنت کردن لاگ اوت داخل یو ار ال، لینک به ویو لاگ اوت داخل تمپلیت، اسم تمپلیت لاگ اوت جنگو و محتویاتشو کپی میکنیم برای اورراید کردن داخل اپ خودمون، روش دوم داخل ستینگ این متغیر رو ست میکنیم `LOGOUT_REDIRECT_URL='account:login'`

## ۱.۴ رفع باگ

اگر نویسنده هیچ مقاله ای نداشت لیست خالی نشون نده بگه کاربر هیچ مقاله ای ندارد با ایف ابجکت لیست و الس و اچ ۳

## ۵ قسمت ۵۵ اکتیو منو و حذف مقاله از بخش ویرایش

استایل سفید افزودن مقاله و بخش های دیگه داخل پنل ادمین، `request.resolver_match` اطلاعات بیشتری از ریکوست میده داخل اون نو باره گوشه کلاس اکتیو میاد سفید میکنه یک شرط با `{request.resolver_match.url_name}` مینویسیم یا اینکه با تمپلیت تگ بنویسیمش (اینو چک کن)

## ۱.۵ حذف مقاله در قسمت ویرایش

داخل تمپلیت یدونه شرط میذاریم که اگر کاربر سوپر یوزر بود و مقاله پی کی داشت (از ریزالور استفاده میکنیم که مطمئن بشیم داخل اپدیت هستیم نه کیریت)

## ۲.۵ چجوری `template tags` های خودمون بسازیم

۱. داخل اپ یدونه دایرکتوری میسازیم به اسم `templatetags`

۲. فایل تمپلیت تگ مثلا `basetags.py` و اینیت داخلش میسازیم

۳. از جنگو تمپلیت رو ایمپورت میکنیم

۴. تمپلیت رو رجیستر میکنیم `register = template.Library()`

۵. متد تمپلیت تگ رو با فایل اچ تی ام الش رجیستر میکنیم `register.inclusion_tag('file.html')`

۶. یک متد مینویسم با پارامترهای `request, link_name, content`

۷. خروجی متد همون ورودی هاش هست به اضافه `link` که اسم لینک هست یک استرینگ فرمت شده مثلا `account:home`

۸. تمپلیت مربوطه رو میسازیم

### ۳.۵ چجوری منوی اکتیو رو فعال کنیم داخل ساید بار

یعنی اگر لیست مقالات انتخاب شد اون سفید باشه اگه افزودن مقاله جدید انتخاب شد اون سفید باشه و اگر رفتیم برای ویرایش هیچ کدام سفید نباشه

راه اول این هست که با استفاده از `if` و `request.resolver_match` پیاده سازیش کنیم مشکل این روش اینکه که کد تمپلیت شلوغ میشه

روش دوم (بهتر) بیایم از تمپلیت تگ ها استفاده کنیم، بدونه تمپلیت تگ تعریف کنیم و بعد از اون استفاده کنیم

## ۶ قسمت ۵۶ ساخت پیشنمایش مقاله های پیشنهادی در پنل مدیریت

۱. یک ویو مثل ارتیکل دیتیل واسش میسازیم بجای سلاگ از پرایمری کی استفاده میکنیم و کوئری رو هم عوض میکنیم

۲. داخل یو آر ال هم ایمپورت میکنیم

۳. یک محدودیت باشه که فقط برای نویسنده و سوپر یوزر قابل نمایش باشه

۴. یک بدج براش داخل هوم درست کنیم

## ۷ قسمت ۵۷ افزایش وضعیت مقالات

حالت `i` مخواستیم از `p` استفاده کنیم یعنی `pendding` ولی چون واسه پابلیش بود دیگه نمیشد این حالتی هست که مقاله رفت برای چک سوپریوزر و دیگه نویسنده نمیتونه ادیتش کنه حالت `b` یعنی `back` حالتی هست که یعنی سوپریوزر چک کرده و ایراد گرفته ازش دوباره مقاله میشه مثل حالت پیشنهادی (میشه طراحی کرد که یک متن هم اونجا بنویسه سوپر یوزر)

۱. اضافه کردن حالت ها به مدل

۲. تصحیح میکسین که کاربر هم بتونه به `b` دسترسی داشته باشه هم `d`

۳. داخل ویو هوم ادمین هم کار مرحله قبل رو تکرار میکنیم ولی اینجا لیست نداریم از فیلتر `make_list` استفاده میکنیم