

Exercise 25: Adding the jokes Table to Django Admin

 10 to 15 minutes

Every new model you create can be managed in Django admin in the same way that users can be managed. You just need to create and register a `ModelAdmin` class for the model.

Open `jokes/admin.py` for editing and write the following code:⁴⁶

Exercise Code 25.1: `jokes/admin.py`

```
1. from django.contrib import admin
2.
3. from .models import Joke
4.
5. @admin.register(Joke)
6. class JokeAdmin(admin.ModelAdmin):
7.     model = Joke
8.     list_display = ['question', 'created', 'updated']
9.
10.    def get_readonly_fields(self, request, obj=None):
11.        if obj: # editing an existing object
12.            return ('created', 'updated')
13.
14.        return ()
```

Things to notice:

1. After importing `admin` from `django.contrib`, you import the `Joke` model, which you then use in the `JokeAdmin` class to set the model being managed in this class.
2. You set `list_display` to specify the fields that should show up in the jokes listing:

<input type="checkbox"/>	QUESTION	CREATED	UPDATED
<input type="checkbox"/>	Why did the chicken cross the road?	Aug. 6, 2020, 4:10 p.m.	Aug. 6, 2020, 4:10 p.m.

The `list_display` attribute is not required. If you don't set it, there will only be one column in the table with the value returned by the model's `__str__()` displayed.

^{46.} **Don't want to type?** Copy from `starter-code/django-admin/jokes_admin.py`.

3. `readonly_fields` holds fields that you want to be able to read but not edit. You could have set this attribute like this:

```
readonly_fields = ('created', 'updated')
```

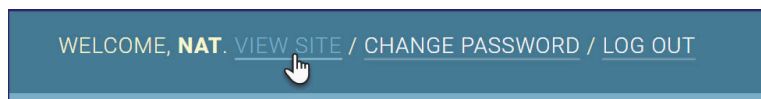
But if you do that, the fields will show up in the form for adding new jokes as well as the form for editing existing jokes. It doesn't make sense that the form for adding new jokes would have readonly fields as those fields will always be empty until the joke is actually added. So, you use the getter function to check if obj exists:

```
def get_readonly_fields(self, request, obj=None):
    if obj: # editing an existing object
        return ('created', 'updated')

    return ()
```

This will tell you if you are editing an existing obj or creating a new one. If you're editing an existing object, you return `('created', 'updated')`. If you're creating a new one, you return an empty tuple, indicating that there are no readonly fields.

Open Django admin and add some jokes. Then, go to the site, which you can get to by clicking the **VIEW SITE** link at the top of Django admin:



You should see the jokes you added.

Git Commit

Commit your code to Git.