

```
1 from django.conf.urls import url
2
3 # Register your models here.
4
```

Settings:



Results:

All clear, no errors found

1
2
3
4
5
6
7

```
from django.apps import AppConfig
class LikedConfig(AppConfig):
    default_auto_field = 'django.db.models.AutoField'
    name = 'likes'
```

Settings:



Results:

All clear, no errors found

```

1 from django.db import models
2 from django.contrib.auth.models import User
3 from posts.models import Post
4
5
6 class Like(models.Model):
7     """
8     Like model, related to "owner" and "post".
9     "owner" is a User instance and "post" is a Post instance.
10    "unique_together" makes sure a user can't like the same post twice.
11    """
12    owner = models.ForeignKey(User, on_delete=models.CASCADE)
13    post = models.ForeignKey(
14        Post, related_name="likes", on_delete=models.CASCADE
15    )
16    created_at = models.DateTimeField(auto_now_add=True)
17
18 class Meta:
19     ordering = ["-created_at"]
20     unique_together = ["owner", "post"]
21
22 def __str__(self):
23     return f'{self.owner} {self.post}'
24

```

Settings:



Results:

All clear, no errors found

```

1 from django.db import IntegrityError
2 from rest_framework import serializers
3 from likes.models import Like
4
5
6 class LikeSerializer(serializers.ModelSerializer):
7     """
8     Serializer for the like model
9     The create method handles the unique constraint on 'owner' and 'post'
10    """
11    owner = serializers.ReadOnlyField(source='owner.username')
12
13    class Meta:
14        model = Like
15        fields = ('id', 'created_at', 'owner', 'post')
16
17    def create(self, validated_data):
18        try:
19            return super().create(validated_data)
20        except IntegrityError:
21            raise serializers.ValidationError({
22                'detail': 'possible duplicate'
23            })
24

```

Settings:



Results:

All clear, no errors found

```
from django.conf import settings
from django.conf import settings

urlpatterns = [
    path('login/', views.login_view),
    path('logout/', views.logout_view),
]
```

Settings:



Results:

2: W292 no newline at end of file

```

1 from rest_framework import generics, permissions
2 from cookbook_api.permissions import IsOwnerOrReadOnly
3 from likes.models import Like
4 from likes.serializers import LikeSerializer
5
6
7 class LikeList(generics.ListCreateAPIView):
8     """
9     List likes or create a like if logged in.
10    """
11    permission_classes = [permissions.IsAuthenticatedOrReadOnly]
12    serializer_class = LikeSerializer
13    queryset = Like.objects.all()
14
15    def perform_create(self, serializer):
16        serializer.save(owner=self.request.user)
17
18
19 class LikeDetail(generics.RetrieveDestroyAPIView):
20     """
21     Retrieve a like or delete it by id if you own it.
22    """
23    permission_classes = [IsOwnerOrReadOnly]
24    serializer_class = LikeSerializer
25    queryset = Like.objects.all()
26

```

Settings:



Results:

All clear, no errors found