

admin.py



CI Python Linter

```
1 from django.contrib import admin
2 from .models import Comment
3
4 admin.site.register(Comment)
5
```

Settings:

Results:

All clear, no errors found

apps.py



CI Python Linter

```
1 from django.apps import AppConfig
2
3
4 class CommentsConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'comments'
7
```

Settings:

Results:

All clear, no errors found

models.py



CI Python Linter

```
1 from django.db import models
2 from django.contrib.auth.models import User
3 from posts.models import Post
4
5
6 class Comment(models.Model):
7     """
8     Comment model, related to User and Post
9     """
10     owner = models.ForeignKey(User, on_delete=models.CASCADE)
11     post = models.ForeignKey(Post, on_delete=models.CASCADE)
12     created_at = models.DateTimeField(auto_now_add=True)
13     updated_at = models.DateTimeField(auto_now=True)
14     content = models.TextField()
15
16     class Meta:
17         ordering = ['-created_at']
18
19     def __str__(self):
20         return self.content
21
```

Settings:

Results:

All clear, no errors found

serializers.py

code  
institute

CI Python Linter

```
15 created_at = serializers.SerializerMethodField()
16 updated_at = serializers.SerializerMethodField()
17
18 def get_is_owner(self, obj):
19     request = self.context['request']
20     return request.user == obj.owner
21
22 def get_comment_like_id(self, obj):
23     user = self.context['request'].user
24     if user.is_authenticated:
25         comment_like = CommentLike.objects.filter(
26             owner=user, comment=obj
27         ).first()
28         return comment_like.id if comment_like else None
29     return None
30
31 def get_created_at(self, obj):
32     return naturaltime(obj.created_at)
33
34 def get_updated_at(self, obj):
35     return naturaltime(obj.updated_at)
36
37 class Meta:
38     model = Comment
39     fields = [
40         'id', 'owner', 'post', 'created_at', 'updated_at', 'content',
41         'is_owner', 'profile_id', 'profile_image', 'comment_like_id',
42         'comment_likes_count'
43     ]
44
45 class CommentDetailSerializer(CommentSerializer):
46     """Serializer for Comment detail view"""
47     post = serializers.ReadOnlyField(source='post.id')
48
49
```

Settings:

Results:

All clear, no errors found

urls.py

code  
institute

CI Python Linter

```
1 from django.urls import path
2 from comments import views
3
4 urlpatterns = [
5     path('comments/', views.CommentList.as_view()),
6     path('comments/<int:pk>/', views.CommentDetail.as_view()),
7 ]
8
```

Settings:

Results:

All clear, no errors found




views.py

code  
institute

CI Python Linter

```
1 from django.db.models import Count
2 from rest_framework import generics, permissions, filters
3 from django_filters.rest_framework import DjangoFilterBackend
4 from def_postcards.permissions import IsOwnerOrReadOnly
5 from .models import Comment
6 from .serializers import CommentSerializer, CommentDetailSerializer
7
8
9 class CommentList(generics.ListCreateAPIView):
10     """List comments or create comments if logged in"""
11     serializer_class = CommentSerializer
12     permission_classes = [permissions.IsAuthenticatedOrReadOnly]
13     queryset = Comment.objects.annotate(
14         comment_likes_count=Count('comment_likes', distinct=True)
15     ).order_by('-created_at')
16     filter_backends = [filters.OrderingFilter, DjangoFilterBackend]
17     filterset_fields = ['post']
18     ordering_fields = [
19         'comment_likes_count',
20         'comment_likes_created_at'
21     ]
22
23 def perform_create(self, serializer):
24     serializer.save(owner=self.request.user)
25
26
27 class CommentDetail(generics.RetrieveUpdateDestroyAPIView):
28     """Retrieve a specific comment, can update or delete if owner"""
29     permission_classes = [IsOwnerOrReadOnly]
30     serializer_class = CommentDetailSerializer
31     queryset = Comment.objects.annotate(
32         comment_likes_count=Count('comment_likes', distinct=True)
33     ).order_by('-created_at')
34
```

Settings:

Results:

All clear, no errors found