

admin.py



CI Python Linter

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

Settings:

Results:

All clear, no errors found

apps.py



CI Python Linter

```
1 from django.apps import AppConfig
2
3
4 class CommentLikesConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'comment_likes'
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 comments.models import Comment
4
5
6 class CommentLike(models.Model):
7     """
8     Comment like model, related to User and Comment
9     Uses 'unique_together' to make sure users can't like the same comment twice
10    """
11     owner = models.ForeignKey(User, on_delete=models.CASCADE)
12     comment = models.ForeignKey(
13         Comment, on_delete=models.CASCADE, related_name='comment_likes'
14     )
15     created_at = models.DateTimeField(auto_now_add=True)
16
17     class Meta:
18         ordering = ['-created_at']
19         unique_together = ['owner', 'comment']
20
21     def __str__(self):
22         return f'{self.owner} likes {self.comment}'
23 |
```


Settings:

Results:

All clear, no errors found




serializers.py



CI Python Linter

```
1 from django.db import IntegrityError
2 from rest_framework import serializers
3 from .models import CommentLike
4
5
6 class CommentLikeSerializer(serializers.ModelSerializer):
7     """
8     Serializer for Comment Like model
9     Create method ensures no duplicates
10    """
11    owner = serializers.ReadOnlyField(source='owner.username')
12
13    class Meta:
14        model = CommentLike
15        fields = [
16            'id', 'created_at', 'owner', 'comment'
17        ]
18
19    def create(self, validated_data):
20        try:
21            return super().create(validated_data)
22        except IntegrityError:
23            raise serializers.ValidationError({'detail': 'possible duplicate'})
24
```

Settings:

Results:

All clear, no errors found

urls.py



CI Python Linter

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

Settings:

Results:

All clear, no errors found




views.py



CI Python Linter

```
1 from rest_framework import generics, permissions
2 from drf_postcards.permissions import IsOwnerOrReadOnly
3 from .models import CommentLike
4 from .serializers import CommentLikeSerializer
5
6
7 class CommentLikeList(generics.ListCreateAPIView):
8     """
9     Lists comment likes
10    Perform create method creates a comment like for logged in users only
11    """
12    permission_classes = [permissions.IsAuthenticatedOrReadOnly]
13    serializer_class = CommentLikeSerializer
14    queryset = CommentLike.objects.all()
15
16    def perform_create(self, serializer):
17        serializer.save(owner=self.request.user)
18
19
20 class CommentLikeDetail(generics.RetrieveDestroyAPIView):
21     """Retrieve all comment likes or delete user's own comment like"""
22    permission_classes = [IsOwnerOrReadOnly]
23    serializer_class = CommentLikeSerializer
24    queryset = CommentLike.objects.all()
25
```

Settings:

Results:

All clear, no errors found