

admin.py

 CI Python Linter

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

Settings:

🌙 ☒ ☀️

---

Results:

All clear, no errors found

apps.py

 CI Python Linter

```
1 from django.apps import AppConfig
2
3
4 class FollowersConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'followers'
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
4
5 class Follower(models.Model):
6     """
7     Follower model, related to User and Post
8     Uses 'unique_together' to prevent duplicate follows by the same user
9     """
10    owner = models.ForeignKey(
11        User, on_delete=models.CASCADE, related_name='following'
12    )
13    followed = models.ForeignKey(
14        User, on_delete=models.CASCADE, related_name='followed'
15    )
16    created_at = models.DateTimeField(auto_now_add=True)
17
18    class Meta:
19        ordering = ['-created_at']
20        unique_together = ['owner', 'followed']
21
22    def __str__(self):
23        return f'{self.owner} follows {self.followed}'
24
```

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 Follower
4
5
6 class FollowerSerializer(serializers.ModelSerializer):
7     """Serializer for Follower model, gets the name of followed user"""
8     owner = serializers.ReadOnlyField(source='owner.username')
9     followed_name = serializers.ReadOnlyField(source='followed.username')
10
11    class Meta:
12        model = Follower
13        fields = ['id', 'created_at', 'owner', 'followed', 'followed_name']
14
15    def create(self, validated_data):
16        try:
17            return super().create(validated_data)
18        except IntegrityError:
19            raise serializers.ValidationError({'detail': 'possible duplicate'})
20
```

Settings:

🌙 ☒ ☀️

---

Results:

All clear, no errors found

urls.py



CI Python Linter

```
1 from django.urls import path
2 from followers import views
3
4 urlpatterns = [
5     path('followers/', views.FollowerList.as_view()),
6     path('followers/<int:pk>/', views.FollowerDetail.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 Follower
4 from .serializers import FollowerSerializer
5
6
7 class FollowerList(generics.ListCreateAPIView):
8     """List user followers or follow a user if logged in"""
9     permission_classes = [permissions.IsAuthenticatedOrReadOnly]
10    serializer_class = FollowerSerializer
11    queryset = Follower.objects.all()
12
13    def perform_create(self, serializer):
14        serializer.save(owner=self.request.user)
15
16
17 class FollowerDetail(generics.RetrieveDestroyAPIView):
18     """Retrieve a follower or unfollow a user"""
19    permission_classes = [IsOwnerOrReadOnly]
20    serializer_class = FollowerSerializer
21    queryset = Follower.objects.all()
22
```

Settings:

Results:

All clear, no errors found