

asgi.py



CI Python Linter

```
1 """
2 ASGI config for drf_postcards project.
3
4 It exposes the ASGI callable as a module-level variable named ``application``.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.2/howto/deployment/asgi/
8 """
9
10 import os
11
12 from django.core.asgi import get_asgi_application
13
14 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'drf_postcards.settings')
15
16 application = get_asgi_application()
17
```

Settings:

Results:

All clear, no errors found

permissions.py



CI Python Linter

```
1 from rest_framework import permissions
2
3
4 class IsOwnerOrReadOnly(permissions.BasePermission):
5     def has_object_permission(self, request, view, obj):
6         if request.method in permissions.SAFE_METHODS:
7             return True
8         return obj.owner == request.user
9
```

Settings:

Results:

All clear, no errors found



serializers.py



CI Python Linter

```
1 from dj_rest_auth.serializers import UserDetailsSerializer
2 from rest_framework import serializers
3
4
5 class CurrentUserSerializer(UserDetailsSerializer):
6     profile_id = serializers.ReadOnlyField(source='profile.id')
7     profile_image = serializers.ReadOnlyField(source='profile.image.url')
8
9 class Meta(UserDetailsSerializer.Meta):
10     fields = UserDetailsSerializer.Meta.fields + (
11         'profile_id', 'profile_image'
12     )
13
```


Settings:

Results:

All clear, no errors found




settings.py



CI Python Linter

```
1 """
2 Django settings for drf_postcards project.
3
4 Generated by 'django-admin startproject' using Django 3.2.23.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.2/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.2/ref/settings/
11 """
12
13 from pathlib import Path
14 import os
15 import dj_database_url
16
17 if os.path.exists('env.py'):
18     import env
19
20 CLOUDINARY_STORAGE = {
21     'CLOUDINARY_URL': os.environ.get('CLOUDINARY_URL')
22 }
23 MEDIA_URL = '/media/'
24 DEFAULT_FILE_STORAGE = 'cloudinary_storage.storage.MediaCloudinaryStorage'
25
26 # Build paths inside the project like this: BASE_DIR / 'subdir'.
27 BASE_DIR = Path(__file__).resolve().parent.parent
28
29 REST_FRAMEWORK = {
30     'DEFAULT_AUTHENTICATION_CLASSES': [(
31         'rest_framework.authentication.SessionAuthentication'
32         if 'DEV' in os.environ
33         else 'dj_rest_auth.jwt_auth.JWTCookieAuthentication'
34     )],
35     'DEFAULT_PAGINATION_CLASS':
```


Settings:

Results:

52: E501 line too long (80 > 79 characters)
163: E501 line too long (91 > 79 characters)
166: E501 line too long (81 > 79 characters)
169: E501 line too long (82 > 79 characters)
172: E501 line too long (83 > 79 characters)




urls.py



CI Python Linter

```
7
8 1. Add an import: from my_app import views
9 2. Add a URL to urlpatterns: path('', views.home, name='home')
10
11 Class-based views
12 1. Add an import: from other_app.views import Home
13 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
14 Including another URLconf
15 1. Import the include() function: from django.urls import include, path
16 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
17
18 """
19 from django.contrib import admin
20 from django.urls import path, include
21 from django.views.generic import TemplateView
22 from .views import logout_route
23
24 urlpatterns = [
25     path('', TemplateView.as_view(template_name='index.html')),
26     path('admin/', admin.site.urls),
27     path('api/api-auth/', include('rest_framework.urls')),
28     path('api/dj-rest-auth/logout/', logout_route),
29     path('api/dj-rest-auth/', include('dj_rest_auth.urls')),
30     path(
31         'api/dj-rest-auth/registration/', include(
32             'dj_rest_auth.registration.urls'
33         )
34     ),
35     path('api/', include('profiles.urls')),
36     path('api/', include('posts.urls')),
37     path('api/', include('comments.urls')),
38     path('api/', include('likes.urls')),
39     path('api/', include('comment_likes.urls')),
40     path('api/', include('followers.urls')),
41     path('api/', include('contact.urls')),
42 ]
43
44 handler404 = TemplateView.as_view(template_name='index.html')
45
```

Settings:

Results:

All clear, no errors found

views.py



CI Python Linter

```
5 JWT_AUTH_SECURE
6 )
7
8
9 @api_view()
10 def root_route(request):
11     return Response({
12         "message": "Welcome to the Postcards API!"
13     })
14
15
16 # dj-rest-auth logout view fix
17 @api_view(['POST'])
18 def logout_route(request):
19     response = Response()
20     response.set_cookie(
21         key=JWT_AUTH_COOKIE,
22         value='',
23         httponly=True,
24         expires=Thu, 01 Jan 1970 00:00:00 GMT,
25         max_age=0,
26         samesite=JWT_AUTH_SAMESITE,
27         secure=JWT_AUTH_SECURE,
28     )
29     response.set_cookie(
30         key=JWT_AUTH_REFRESH_COOKIE,
31         value='',
32         httponly=True,
33         expires=Thu, 01 Jan 1970 00:00:00 GMT,
34         max_age=0,
35         samesite=JWT_AUTH_SAMESITE,
36         secure=JWT_AUTH_SECURE,
37     )
38     return response
39
```

Settings:

Results:

All clear, no errors found



wsgi.py



CI Python Linter

```
1 """
2 WSGI config for drf_postcards project.
3
4 It exposes the WSGI callable as a module-level variable named ``application``.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.2/howto/deployment/wsgi/
8 """
9
10 import os
11
12 from django.core.wsgi import get_wsgi_application
13
14 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'drf_postcards.settings')
15
16 application = get_wsgi_application()
17
```

Settings:

Results:

All clear, no errors found