QMS Backend

~~~
```python

##myproject

#api

#migartions
#0001_initial.py

# Generated by Django 5.1.1 on 2024-09-30 06:29

from django.db import migrations, models


class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Item',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('name', models.CharField(max_length=100)),
                ('category', models.CharField(max_length=50)),
                ('received_date', models.DateTimeField(auto_now_add=True)),
                ('quality_status', models.CharField(max_length=50)),
                ('rejected_reason', models.TextField(blank=True, null=True)),
            ],
        ),
    ]



#apps.py

from django.apps import AppConfig


class ApiConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'api'
~~~

```
#models.py

# Create your models here.
from django.db import models

class Item(models.Model):
    name = models.CharField(max_length=100)
    category = models.CharField(max_length=50)
    received_date = models.DateTimeField(auto_now_add=True)
    quality_status = models.CharField(max_length=50)
    rejected_reason = models.TextField(null=True, blank=True)

    def _str_(self):  # Fixed method name
        return self.name

#Serializer.py

from rest_framework import serializers
from .models import Item

class ItemSerializer(serializers.ModelSerializer):
    class Meta:
        model = Item
        fields = '__all__'  # Use 'all_' to include all fields, or specify a list of fields

#test.py

from django.test import TestCase

# Create your tests here.

#urls.py

# from django.contrib import admin
# from django.urls import path, include
# from rest_framework_simplejwt.views import (
#     TokenObtainPairView,
#     TokenRefreshView,
# )

# urlpatterns = [
#     path('admin/', admin.site.urls),
#     path('api/', include('api.urls')),  # Include the routes from api/urls.py

#     # JWT authentication endpoints
#     path('api/token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
#     path('api/token/refresh/', TokenRefreshView.as_view(), name='token_refresh'),
# ]

from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import ItemViewSet

router = DefaultRouter()
router.register(r'items', ItemViewSet)

urlpatterns = [
    path('', include(router.urls)),  # Correct, no recursion or leading slash
]

#views.py

# Create your views here.
# from rest_framework import viewsets
# from .models import Item
# from .serializers import ItemSerializer

# class ItemViewSet(viewsets.ModelViewSet):
#     queryset = Item.objects.all()
#     serializer_class = ItemSerializer

from rest_framework import viewsets
from .models import Item
from .serializers import ItemSerializer

class ItemViewSet(viewsets.ModelViewSet):
    queryset = Item.objects.all()
    serializer_class = ItemSerializer

#myproject

 #asgi.py

    """
ASGI config for myproject project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/5.1/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'myproject.settings')

application = get_asgi_application()

#settings.py

"""
Django settings for myproject project.
```

```
Generated by 'django-admin startproject' using Django 5.1.1.

For more information on this file, see
https://docs.djangoproject.com/en/5.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/5.1/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure--*oq8-$$wa%^lx6%^j24=fxd#@1hp(=le-8jkw)32_g6-#se3!'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    # other installed apps
    'api',
    'rest_framework',  # Add this if you're using Django REST Framework

]
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
        'rest_framework_simplejwt.authentication.JWTAuthentication',
    ),
}
from datetime import timedelta

SIMPLE_JWT = {
    'ACCESS_TOKEN_LIFETIME': timedelta(minutes=30),
    'REFRESH_TOKEN_LIFETIME': timedelta(days=1),
    'ROTATE_REFRESH_TOKENS': False,
    'BLACKLIST_AFTER_ROTATION': True,
}

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'myproject.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'myproject.wsgi.application'


# Database
# https://docs.djangoproject.com/en/5.1/ref/settings/#databases

# DATABASES = {
#     'default': {
#         'ENGINE': 'django.db.backends.sqlite3',
#         'NAME': BASE_DIR / 'db.sqlite3',
#     }
# }

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'hospital_quality',        # Replace with your MySQL database name
        'USER': 'root',        # Replace with your MySQL username
        'PASSWORD': 'Volvoxc40@p11',  # Replace with your MySQL password
        'HOST': 'localhost',
        'PORT': '3306',  # Default MySQL port
    }
}
```

```
APPEND_SLASH = True


# Password validation
# https://docs.djangoproject.com/en/5.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/5.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.1/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/5.1/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

#urls.py

"""
URL configuration for myproject project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/5.1/topics/http/urls/
```

```
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
# from django.contrib import admin
# from django.urls import path

# urlpatterns = [
#     path('admin/', admin.site.urls),
# ]

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
   path('api/', include('api.urls')),  # yourapp should be replaced with your app name
]

#wsgi.py

"""
WSGI config for myproject project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/5.1/howto/deployment/wsgi/
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'myproject.settings')

application = get_wsgi_application()

#manage.py

#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys
```

```
def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'myproject.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```