1. *Testing the built API's :*

You can use pytest as an alternative to Django's built-in test runnerInstall it in the main root directory:

```
pip install pytest
```

And also modify this in setting.py in INSTALLED APPS :

```
INSTALLED_APPS = [
    'django_filters',
]
```

After installing these packages, you should have everything you need to write and run test cases for your Django APIs.

Your directory should be like this :

```
BasicApp/
├── coins/
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── migrations/
│   │   └── __init__.py
│   ├── models.py
│   ├── serializers.py
│   ├── tests.py                    <<test file should be placed here
│   └── views.py
├── manage.py
└── BasicProject/
    ├── __init__.py
    ├── asgi.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
└── media/
    └── coin_images/
```

Once you've placed the test file in the correct directory, you can run the tests again to verify their functionality.

Your test file can be like this to run test case on the functions like list, retrieve, create, update, delete : (tests.py)

```
from django.test import TestCase
from django.urls import reverse
from rest_framework import status
from rest_framework.test import APIClient
from django.core.files.uploadedfile import SimpleUploadedFile
from .models import Coin
class CoinAPITestCase(TestCase):
    def setUp(self):
        self.client = APIClient()
        self.coin1 = Coin.objects.create(
            coin_name='Test Coin 1',
            coin_desc='Description of Test Coin 1',
            coin_year=2022,
            coin_country='Test Country',
            coin_material='Test Material',
            coin_weight=10.5,
            starting_bid=100.0,
            coin_status='available'
        )
    def test_list_coins(self):
        url = reverse('coin-list')
        response = self.client.get(url)
        self.assertEqual(response.status_code, status.HTTP_200_OK)
    def test_retrieve_coin(self):
```

```
        url = reverse('coin-detail', kwargs={'pk': self.coin1.pk})
        response = self.client.get(url)
        self.assertEqual(response.status_code, status.HTTP_200_OK)
    def test_delete_coin(self):
        url = reverse('coin-detail', kwargs={'pk': self.coin1.pk})
        response = self.client.delete(url)
        self.assertEqual(response.status_code, status.HTTP_204_NO_CONTENT)
    def test_create_coin(self):
        url = reverse('coin-list')
        data = {
            'coin_name': 'New Test Coin',
            'coin_desc': 'Description of New Test Coin',
            'coin_year': 2023,
            'coin_country': 'New Test Country',
            'coin_material': 'New Test Material',
            'coin_weight': 15.0,
            'starting_bid': 150.0,
            'coin_status': 'available'
        }
        response = self.client.post(url, data, format='json')
        self.assertEqual(response.status_code, status.HTTP_201_CREATED)


    def test_update_coin(self):
        url = reverse('coin-detail', kwargs={'pk': self.coin1.pk})
        data = {
            'coin_name': 'Updated Test Coin',
            'coin_desc': 'Updated Description of Test Coin 1',
            'coin_year': 2023,
```

```
            'coin_country': 'Updated Test Country',
            'coin_material': 'Updated Test Material',
            'coin_weight': 15.0,
            'starting_bid': 150.0,
            'coin_status': 'available'
        }
        response = self.client.put(url, data, format='json')
        self.assertEqual(response.status_code, status.HTTP_200_OK)
```

After configuring the test file, after activating your project, you can test your cases by running the command :

```
python manage.py test
```

Your expected output in the terminal after testing should be like :

```
(myenv) shyam@HP-Pavilion-Laptop-15-cs1xxx:~/Public/Django/BasicApp$ python manage.py test
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
----------------------------------------------------------------------
Ran 5 tests in 0.017s

OK
Destroying test database for alias 'default'...
(myenv) shyam@HP-Pavilion-Laptop-15-cs1xxx:~/Public/Django/BasicApp$
```

You can see the 5 test cases have been passed with ‘OK’.

==========