Topic: 12. Nested Serializers for Related Models Part 1

Speaker: Personal | Notebook: API Development using Django Framework



For additional reference, use <u>RestAPI Documentation</u> on nested serializer.

You can use the Online JSON Viewer to see/view the code or text.

1. Create the new app BLOGS where each blog post may have several comments. In the terminal, use the create startapp command:

(env)
rosil@LearnCodeRepeat MINGW64 /c/Users/rosil/OneDrive/Documents/MyCodingCareer/Django Projects/API De
REST\_APIProject
\$ python manage.py startapp blogs

2. Register this new app in the SETTINGS.PY:

```
DjangoREST_APIProject
    File
         Edit
               Selection
                         View
        EXPLORER
                                                 models.py employees
ф
                                🕏 settings.py 🗙
                                                                            admin.py students

✓ DJANGOREST_APIPROJECT

                                django_rest_main > 🕏 settings.py > ...
        > api
                                       DEBUG = True

✓ blogs

                                        ALLOWED_HOSTS = []
         > migrations
        __init__.py
        🕏 admin.py
                                        # Application definition
        apps.py
        models.py
                                        INSTALLED APPS = [
                                            'django.contrib.admin',
        tests.py
                                            'django.contrib.auth',
        🕏 views.py
                                            'django.contrib.contenttypes',
Д

✓ django_rest_main

                                            'django.contrib.sessions',
         > __pycache__
                                            'django.contrib.messages',
        🕏 __init__.py
                                             'django.contrib.staticfiles',
        🕏 asgi.py
                                            'rest framework',
        🕏 settings.py
                                            'students',
                                             'employees',
                                  42
        urls.py
                                            'api'.
        wsgi.py
                                  44
                                            'blogs',
        > employees
        > env
        > students

■ db.sqlite3

                                        MIDDLEWARE =
                                             'django.middleware.security.SecurityMiddleware',
       manage.py
                                 PROBLEMS
                                            OUTPUT
                                                     DEBUG CONSOLE
                                                                     TERMINAL
                                                                                PORTS
```

3. Create the models BLOG POST AND COMMENTS:

```
🔀 File Edit Selection View
                                                                                                                      EXPLORER
                              e settings.py
                                              models.py blogs X
                                                                                            admin.py students
                                                                                                                  e urls.py djang
凸

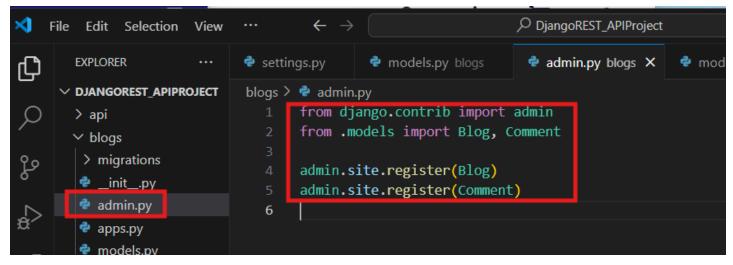
✓ DJANGOREST_APIPROJECT

                              blogs > 🕏 models.py > ..
                                    from django.db import models
       > api

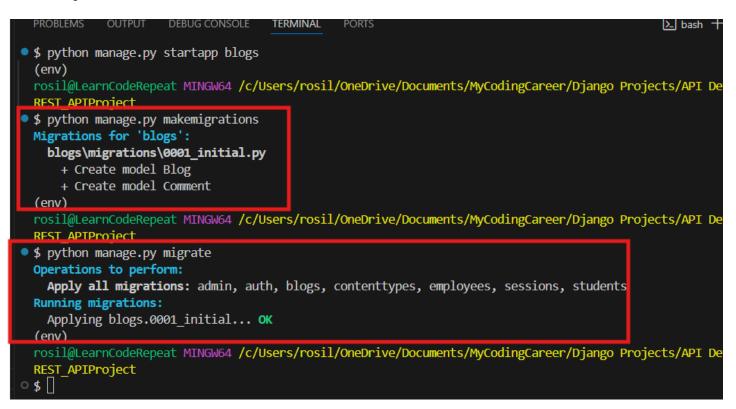
✓ blogs

                                     class Blog(models.Model):
        > migrations
                                         blog_title = models.CharField(max_length=100)
        __init__.py
                                         blog_body = models.TextField()
        admin.py
        apps.py
                                         def __str__(self):
                                             return self.blog_title
       models.py
        tests.py
        views.py
                                     class Comment(models.Model):
       ∨ django_rest_main
                                         blog = models.ForeignKey(Blog, on_delete=models.CASCADE,related_name='comments'
        > _pycache_
                                         comment = models.TextField()
        __init__.py
                                         def __str__(self):
        asgi.py
                                             return self.comment
        settings.py
```

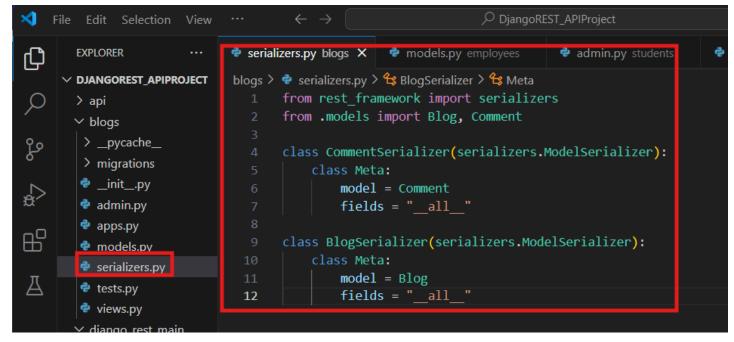
4. Register the models in ADMIN.PY:



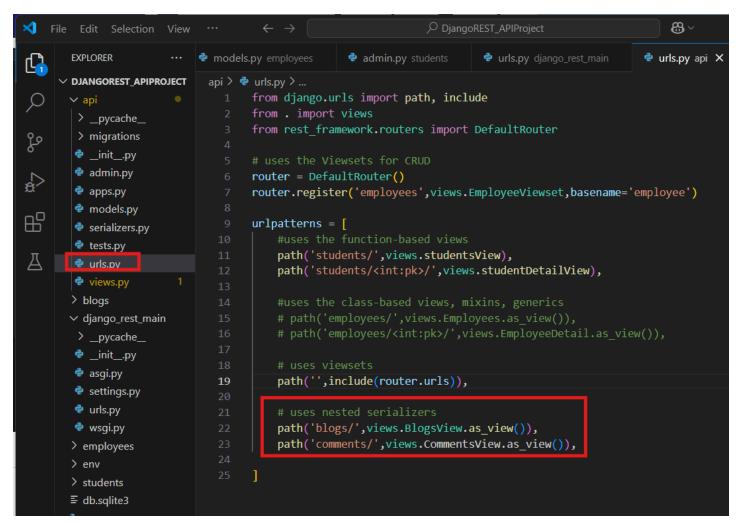
5. Make the migrations.



- 6. Use the admin dashboard to add a new record.
- 7. Create a new file SERIALIZER in the BLOG FOLDER. You may also use the serializer.py in the API folder. This may work so add the serializers.py where you want it. For organization, have it in the app you want to use the serializer with.

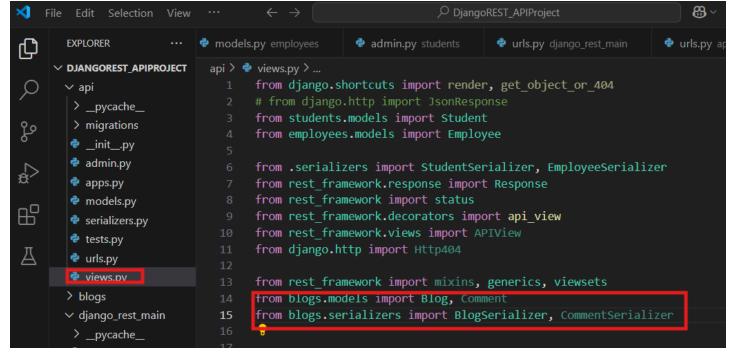


8. Create the path in the API\URLS.PY:

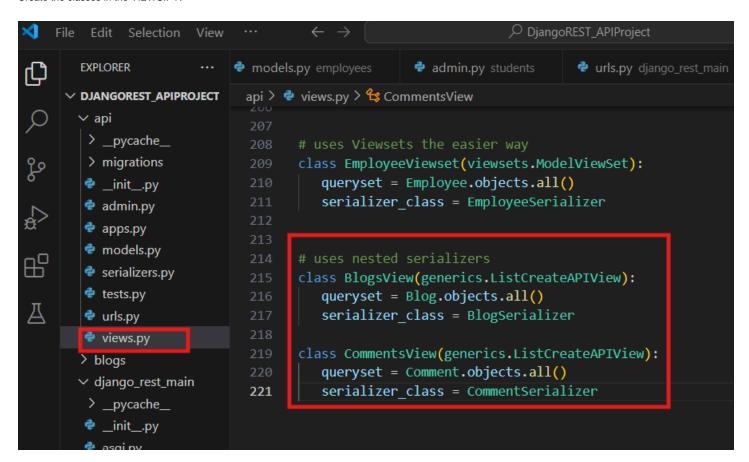


9. Update the VIEWS.PY.

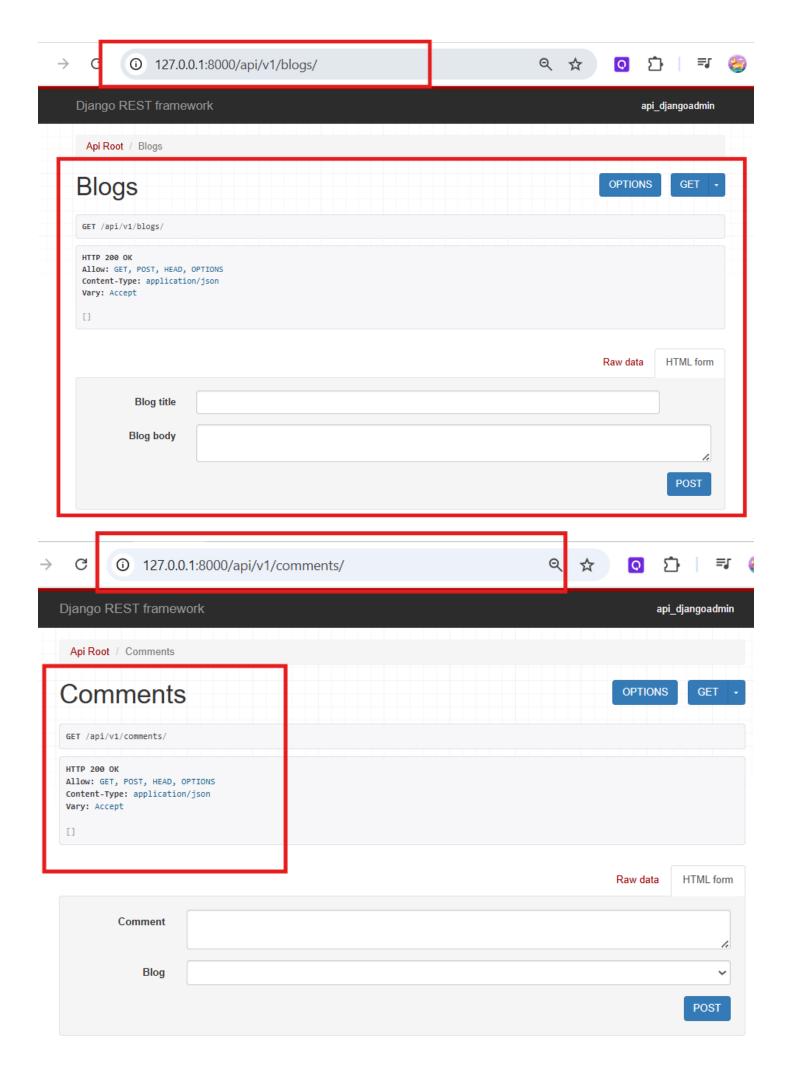
Add the models BLOG & COMMENT:

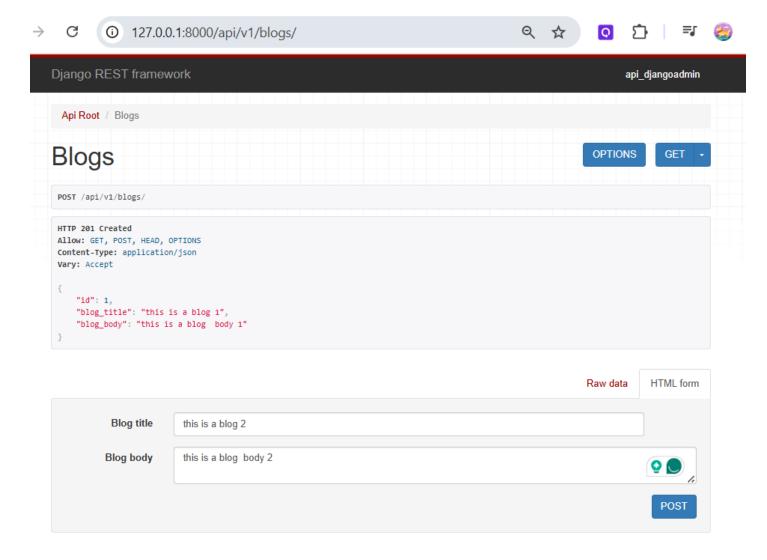


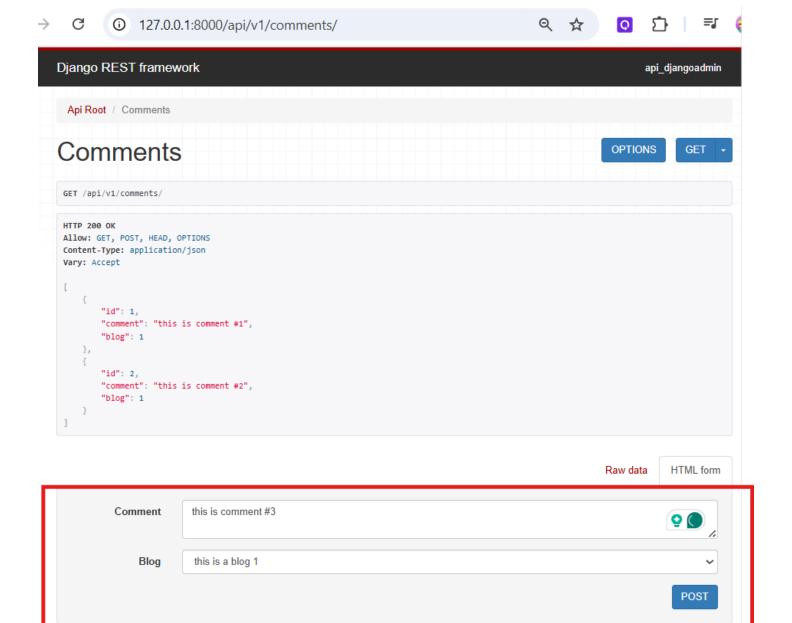
Create the classes in the VIEWS.PY:



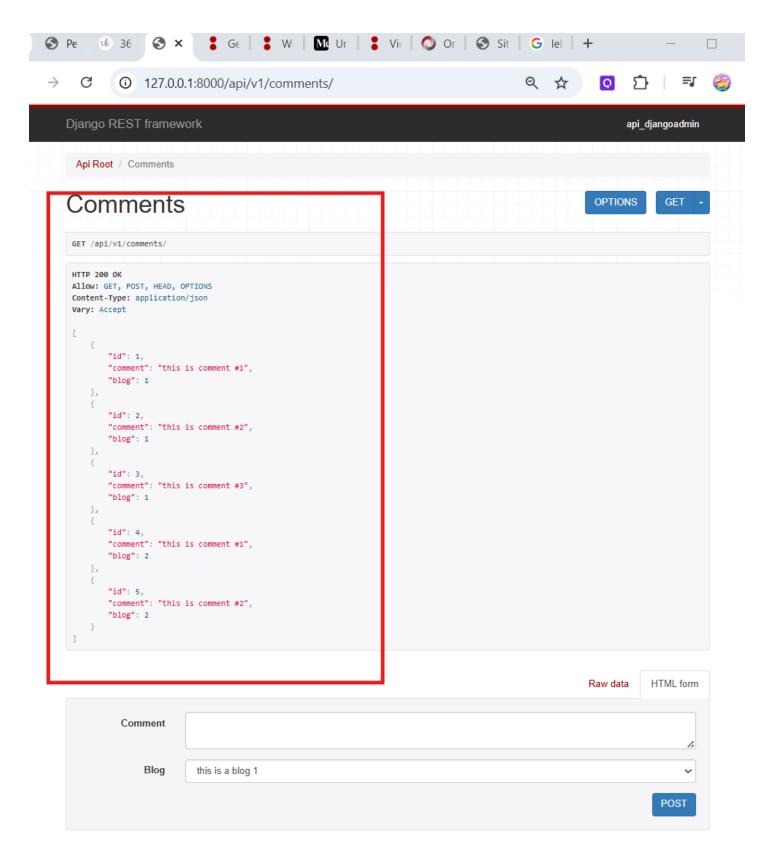
10. View the BLOG and COMMENT models:







12. Our COMMENTS model shall be:



Our BLOGS shall be:

## Django REST framework

```
Api Root / Blogs
Blogs
 GET /api/v1/blogs/
 HTTP 200 OK
 Allow: GET, POST, HEAD, OPTIONS
 Content-Type: application/json
 Vary: Accept
        "id": 1,
        "blog_title": "this is a blog 1",
        "blog_body": "this is a blog body 1"
        "id": 2,
        "blog_title": "this is a blog 2",
        "blog_body": "this is a blog body 2"
```

```
Blog title
Blog body
```

13. To also show the comments under BLOGS model, update the SERIALIZERS.PY:

```
×
       EXPLORER
                               models.py blogs
                                                    admin.py blogs
                                                                         serializers.py blogs •
                                                                                                models.py

✓ DJANGOREST_APIPROJECT

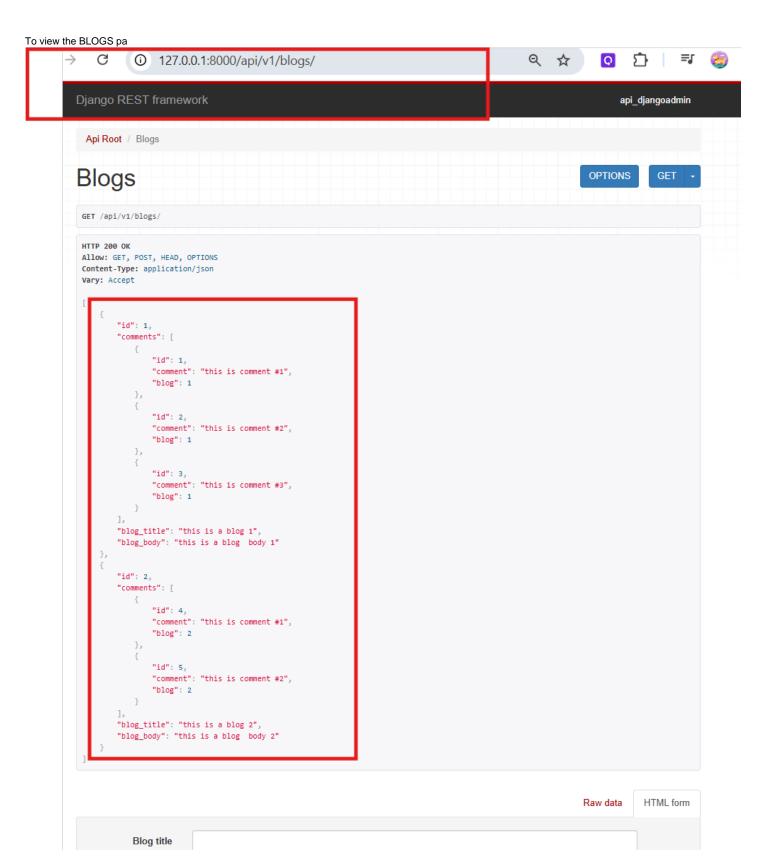
                               blogs > 🕏 serializers.py > ધ BlogSerializer > ધ Meta
                                      from rest framework import serializers

✓ api

                                      from .models import Blog, Comment
         > __pycache__
         > migrations
                                      class CommentSerializer(serializers.ModelSerializer):
        __init__.py
                                          class Meta:
        admin.py
                                              model = Comment
        apps.py
                                               fields = "__all_ "
        models.py
                                      class BlogSerializer(serializers.ModelSerializer):
        serializers.py
                                          # comments is the related name in MODELS.PY
        tests.py
                                          comments = CommentSerializer(many=True, read_only=True)
Д
        urls.py
        views.py
                                          class Meta:

✓ blogs

                                              model = Blog
                                               fields = "__all__"
         > _pycache_
                                15
         > migrations
```

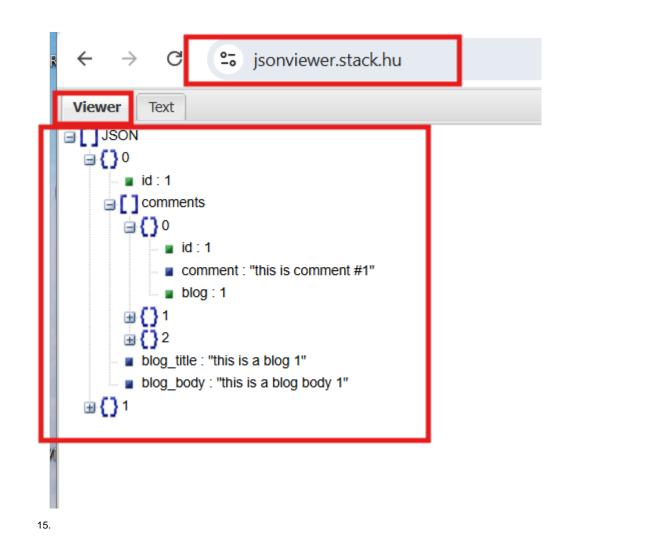


14. So now, copy and paste the sample records to ONLINE JSON VIEWER:

**TEXT MODE:** 

## ← → C % jsonviewer.stack.hu

```
Viewer
       Text
Paste Copy | Format Remove white space | Clear | Load JSON data
   {
        "id": 1,
        "comments": [
                "id": 1,
                "comment": "this is comment #1",
                "blog": 1
            },
                "id": 2,
                "comment": "this is comment #2",
                "blog": 1
            },
                "id": 3,
                "comment": "this is comment #3",
                "blog": 1
        ],
        "blog_title": "this is a blog 1",
        "blog_body": "this is a blog body 1"
   },
{
        "id": 2,
        "comments": [
            {
                "id": 4,
                "comment": "this is comment #1",
                "blog": 2
            },
                "id": 5,
                "comment": "this is comment #2",
                "blog": 2
            }
        "blog_title": "this is a blog 2",
        "blog_body": "this is a blog body 2"
   }
]
```



Copyright © Personal Digital Notebooks | By Rosilie | Date Printed: Dec. 14, 2025, 12:48 p.m.