Topic: 12. Nested Serializers for Related Models Part 1

Speaker: Personal | Notebook: API Development using Django Framework

---

For additional reference, use RestAPI Documentation 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:

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

3. Create the models BLOG POST AND COMMENTS:

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:

Django REST framework

Api Root / Blogs

Blogs

OPTIONS   GET

```
GET /api/v1/blogs/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[]
```

Raw data   HTML form

Blog title

Blog body

POST

Django REST framework

Api Root / Comments

Comments

OPTIONS   GET

```
GET /api/v1/comments/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[]
```

Raw data   HTML form

Comment

Blog

POST

11. Add new records.

Django REST framework   api_djangoadmin

Api Root / Blogs

Blogs   OPTIONS   GET ▾

POST /api/v1/blogs/

HTTP 201 Created
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"
}

                                              Raw data   HTML form

Blog title   this is a blog 2

Blog body   this is a blog body 2

                                                              POST

Django REST framework

api_djangoadmin

Api Root / Comments

Comments

OPTIONS   GET ▾

GET /api/v1/comments/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "id": 1,
        "comment": "this is comment #1",
        "blog": 1
    },
    {
        "id": 2,
        "comment": "this is comment #2",
        "blog": 1
    }
]
```

Raw data | HTML form

Comment | this is comment #3

Blog | this is a blog 1

POST

12. Our COMMENTS model shall be:

127.0.0.1:8000/api/v1/comments/

Api Root / Comments

Comments

OPTIONS   GET ▾

```
GET /api/v1/comments/
```

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
    {
        "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
    },
    {
        "id": 4,
        "comment": "this is comment #1",
        "blog": 2
    },
    {
        "id": 5,
        "comment": "this is comment #2",
        "blog": 2
    }
]
```

Raw data   HTML form

Comment

Blog   this is a blog 1   ⌄

POST

Our BLOGS shall be:

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


```python
from rest_framework import serializers
from .models import Blog, Comment

class CommentSerializer(serializers.ModelSerializer):
    class Meta:
        model = Comment
        fields = "__all__"

class BlogSerializer(serializers.ModelSerializer):
    # comments is the related_name in MODELS.PY
    comments = CommentSerializer(many=True, read_only=True)

    class Meta:
        model = Blog
        fields = "__all__"
```

To view the BLOGS pa

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

TEXT MODE:

jsonviewer.stack.hu

Viewer | **Text**

Paste Copy | Format Remove white space | Clear | Load JSON data

```json
[
    {
        "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"
    }
]
```
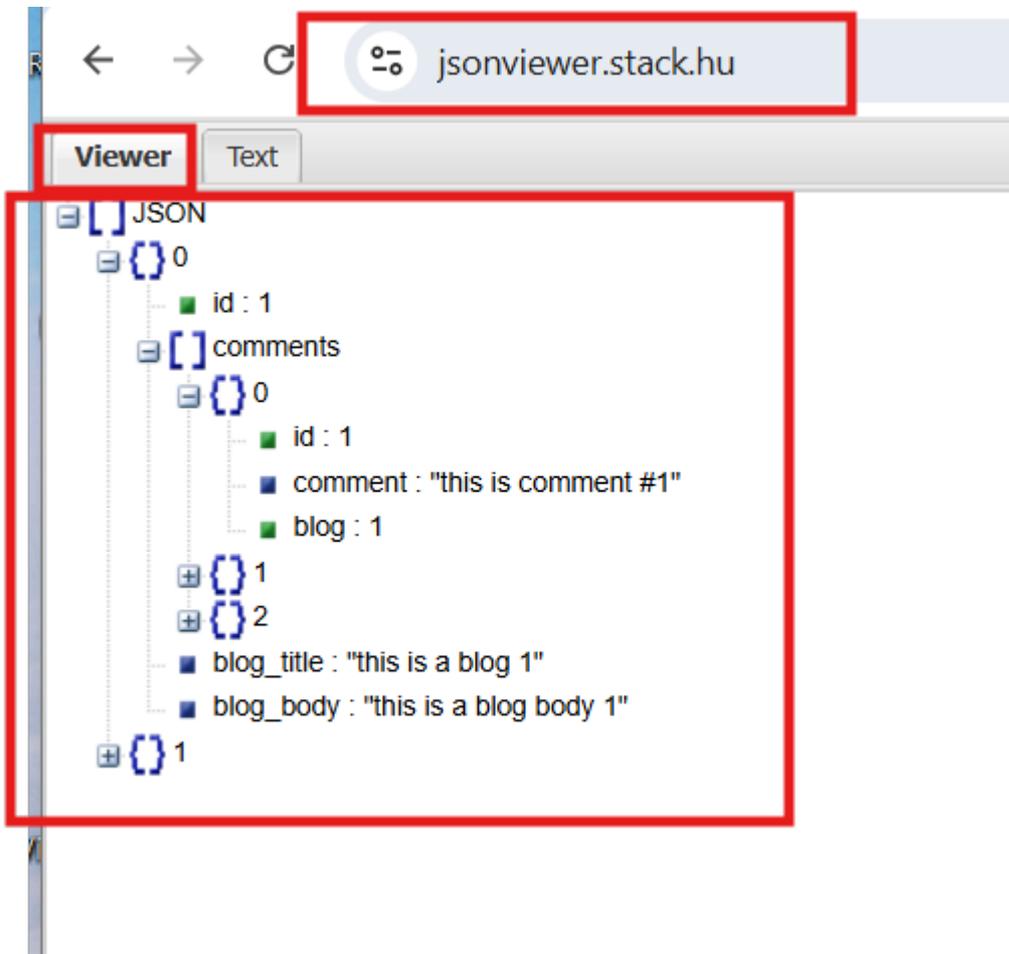
VIEWER MODE:

15.