# Topic: 13. Nested Serializers for Related Models Part 2

*Speaker: Personal | Notebook: API Development using Django Framework*

---



In this post, we use the primary key to allow other CRUD operations.

1. So we update the API\URLS.PY to include the PK-based operations.



2. Update the API\VIEWS.PY

Then update as:



3. Now view the specific path of blog post # 1:

Django REST framework                                            api_djangoadmin

Api Root / Blogs / Blog Detail

# Blog Detail                                    DELETE    OPTIONS    GET  ▾

GET /api/v1/blogs/1/

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

{
    "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"
}
```

                                                    Raw data    HTML form

Blog title     | this is a blog 1                                    |

Blog body      | this is a blog  body 1                              |

                                                                    PUT

Now, you can update and delete this record.

You can also add a new post here:

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

[
    {
        "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"
    }
]
```

Raw data    HTML form

**Blog title**    this is a blog 3

**Blog body**    This is a blog description

POST

Then you can reload your page:

**Django REST framework**                                    **api_djangoadmin**

Api Root / Blogs

# Blogs                                          OPTIONS    GET ▾

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

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

[
    {
        "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"
    },
    {
        "id": 3,
        "comments": [],
        "blog_title": "this is a blog 3",
        "blog_body": "This is a blog description"
    }
]
```

Raw data    HTML form

| Blog title | |
| --- | --- |
| Blog body | |

POST

4. To allow CRUD operations on `Comment` model, then we update the `class CommentDetailView`:

5. To view the COMMENTS path:

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

Raw data    HTML form

| Comment | This is a comment #1 |
|---------|----------------------|
| Blog | this is a blog 3 |

POST

6. To delete the specific comment:

Django REST framework

api_djangoadmin

**Api Root** / **Comments** / Comment Detail

# Comment Detail

DELETE    OPTIONS    GET ▾

```
GET /api/v1/comments/7/
```

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

{
    "id": 7,
    "comment": "test comment",
    "blog": 3
}
```

Raw data    HTML form

| | |
|---|---|
| **Comment** | test comment |
| **Blog** | this is a blog 3 |

PUT

7. View the deleted record:

8.