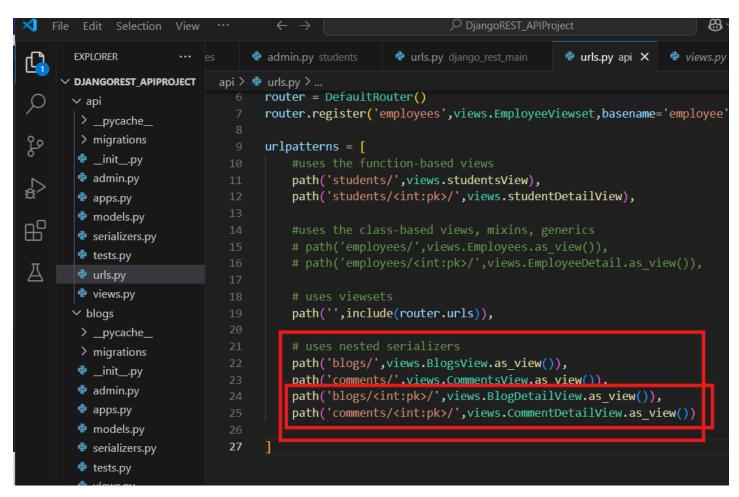
Topic: 13. Nested Serializers for Related Models Part 2

Speaker: Personal | Notebook: API Development using Django Framework

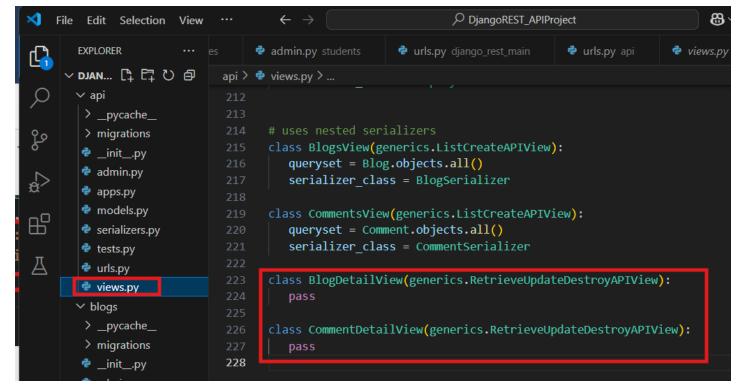


In this post, we use the primary key to allow other CRUD operations.

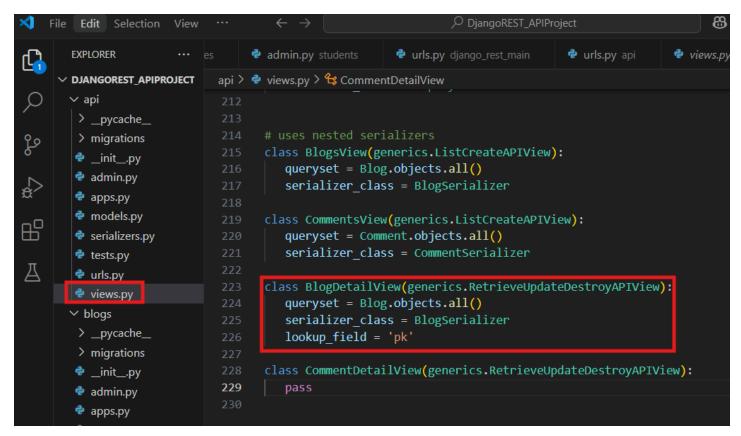
1. So we update the API\URLS.PY to include the PK-based operations.



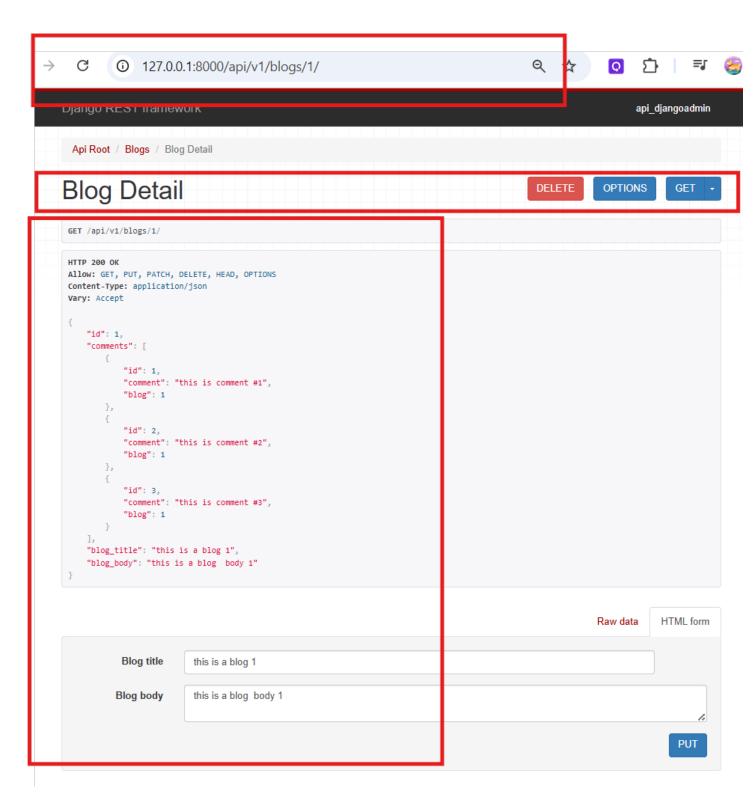
2. Update the API\VIEWS.PY



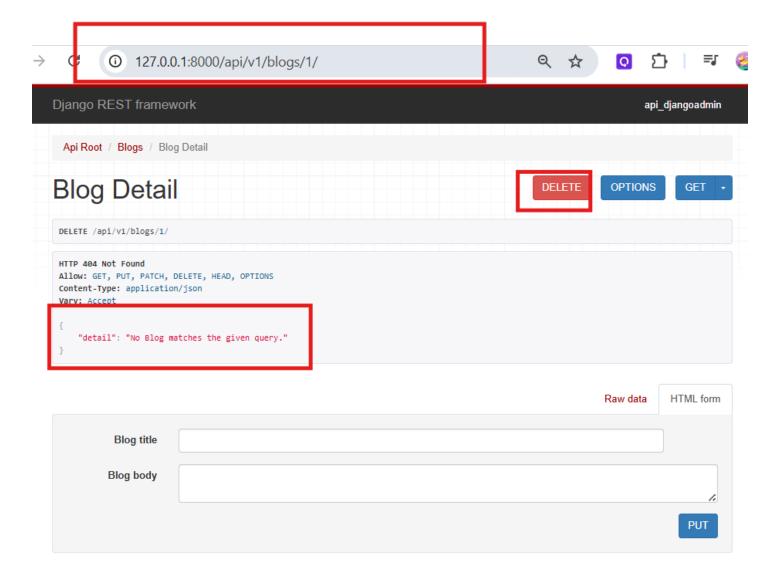
Then update as:



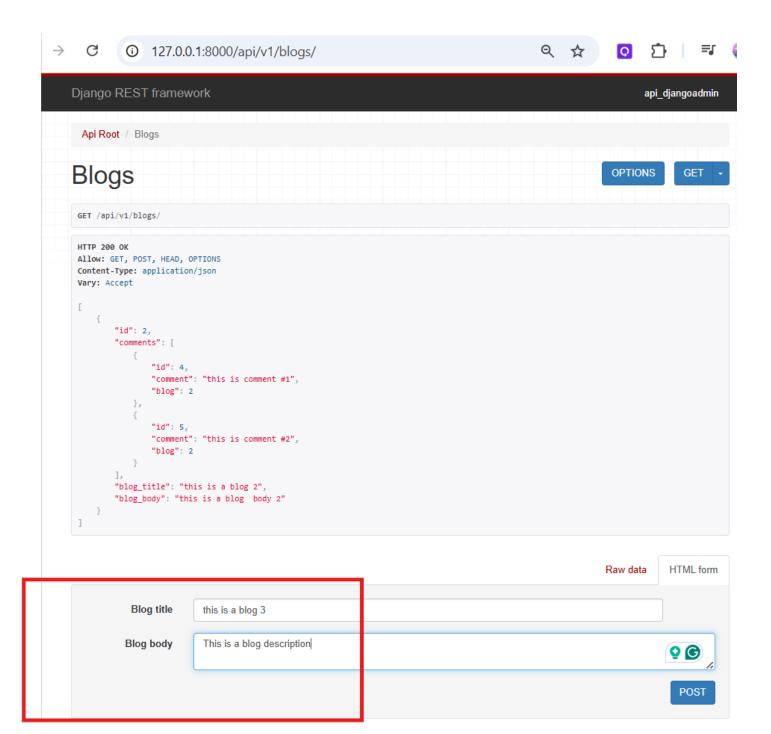
3. Now view the specific path of blog post # 1:



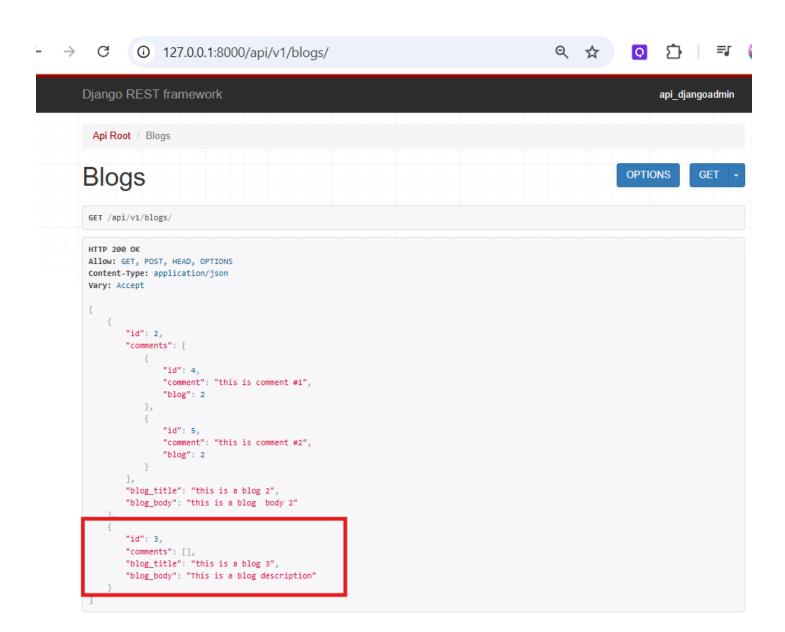
Now, you can update and delete this record.



You can also add a new post here:



Then you can reload your page:



Blog title
Blog body
POST

Raw data

HTML form

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

```
File
        Edit Selection
                       View
                                                                  DjangoREST_APIProject
                                                                                                          88
       EXPLORER
                                   admin.py students
                                                         django_rest_main
                                                                                    🕏 urls.py api
                                                                                                    🕏 views.py

✓ DJANGOREST_APIPROJECT

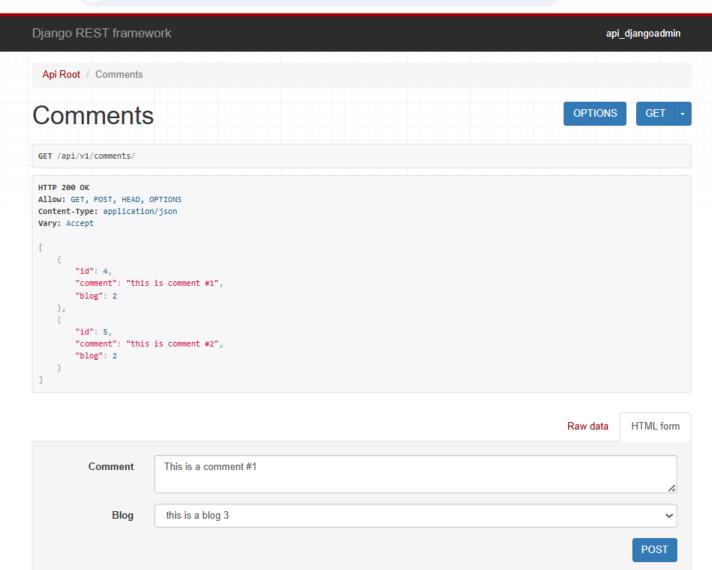
                              api > 🕏 views.py > 😭 CommentDetailView

✓ api

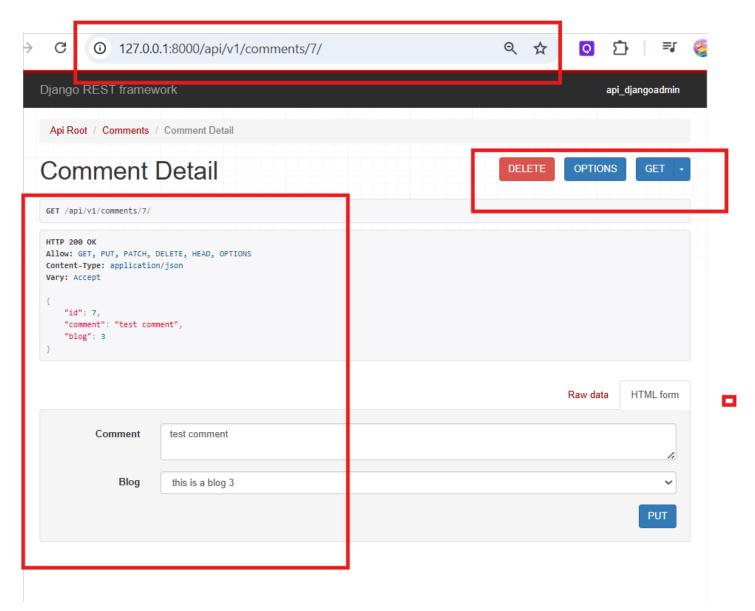
                                     # uses nested serializers
        > _pycache_
                                     class BlogsView(generics.ListCreateAPIView):
        > migrations
                                        queryset = Blog.objects.all()
        __init__.py
                                        serializer_class = BlogSerializer
        admin.py
                                     class CommentsView(generics.ListCreateAPIView):
        apps.py
                                        queryset = Comment.objects.all()
        models.py
品
                                        serializer_class = CommentSerializer
        serializers.py
        tests.py
                                     class BlogDetailView(generics.RetrieveUpdateDestroyAPIView):
Д
        urls.py
                                        queryset = Blog.objects.all()
       🕏 views.py
                                        serializer_class = BlogSerializer
       ∨ blogs
                                        lookup field = 'pk'
        > _pycache_
                                     class CommentDetailView(generics.RetrieveUpdateDestroyAPIView):
        > migrations
                                        queryset = Comment.objects.all()
        __init__.py
                              230
                                        serializer_class = CommentSerializer
        admin.py
                                        lookup_field = 'pk'
        apps.py
        models.py
```

5. To view the COMMENTS path:

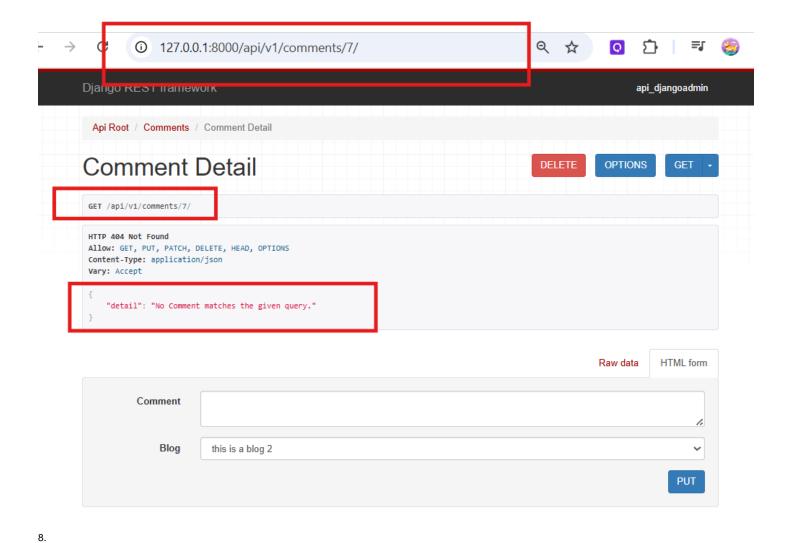




6. To delete the specific comment:



7. View the deleted record:



Copyright © Personal Digital Notebooks | By Rosilie | Date Printed: Dec. 14, 2025, 1:10 p.m.