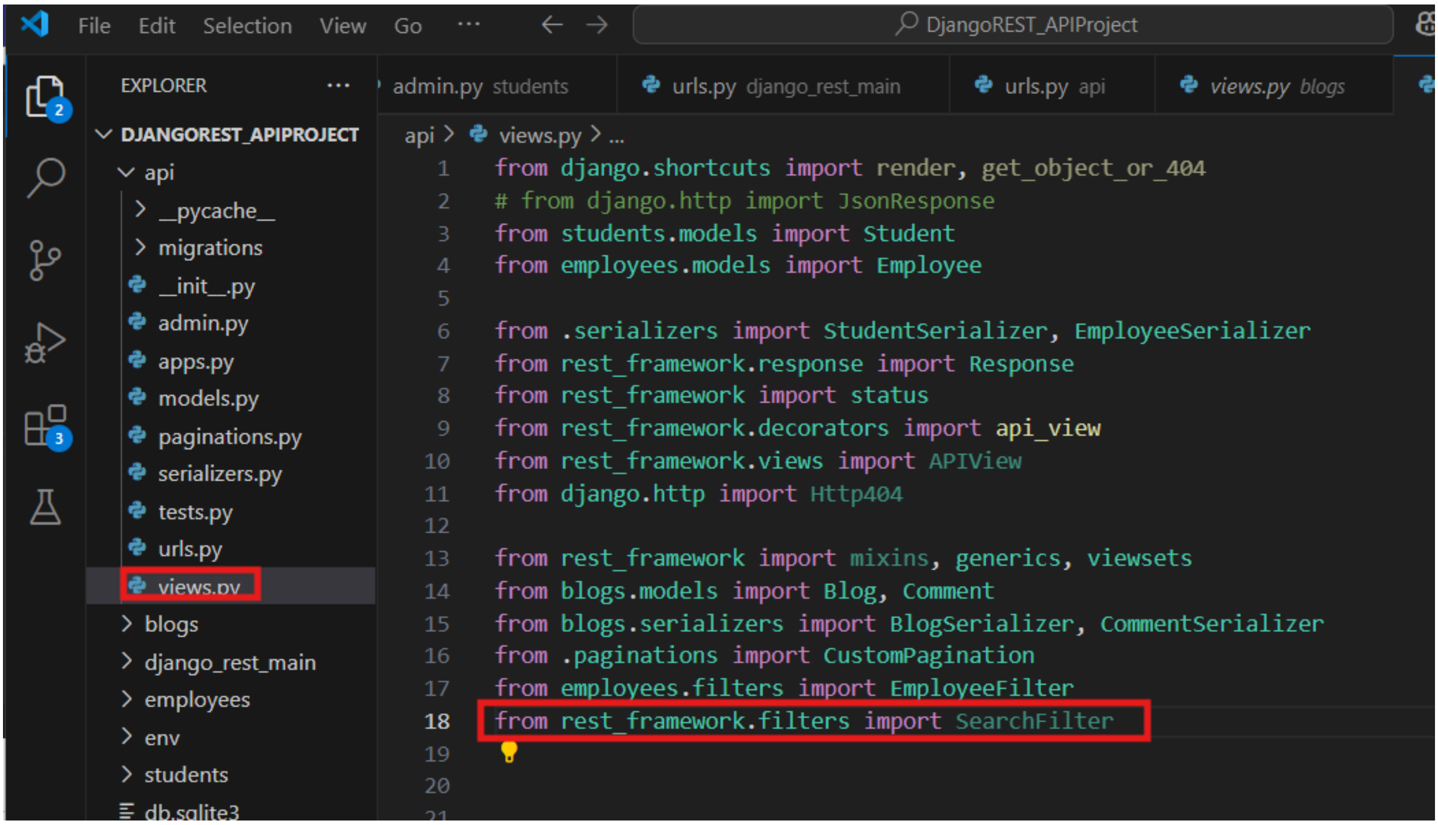
Topic: 16. DRF Searching

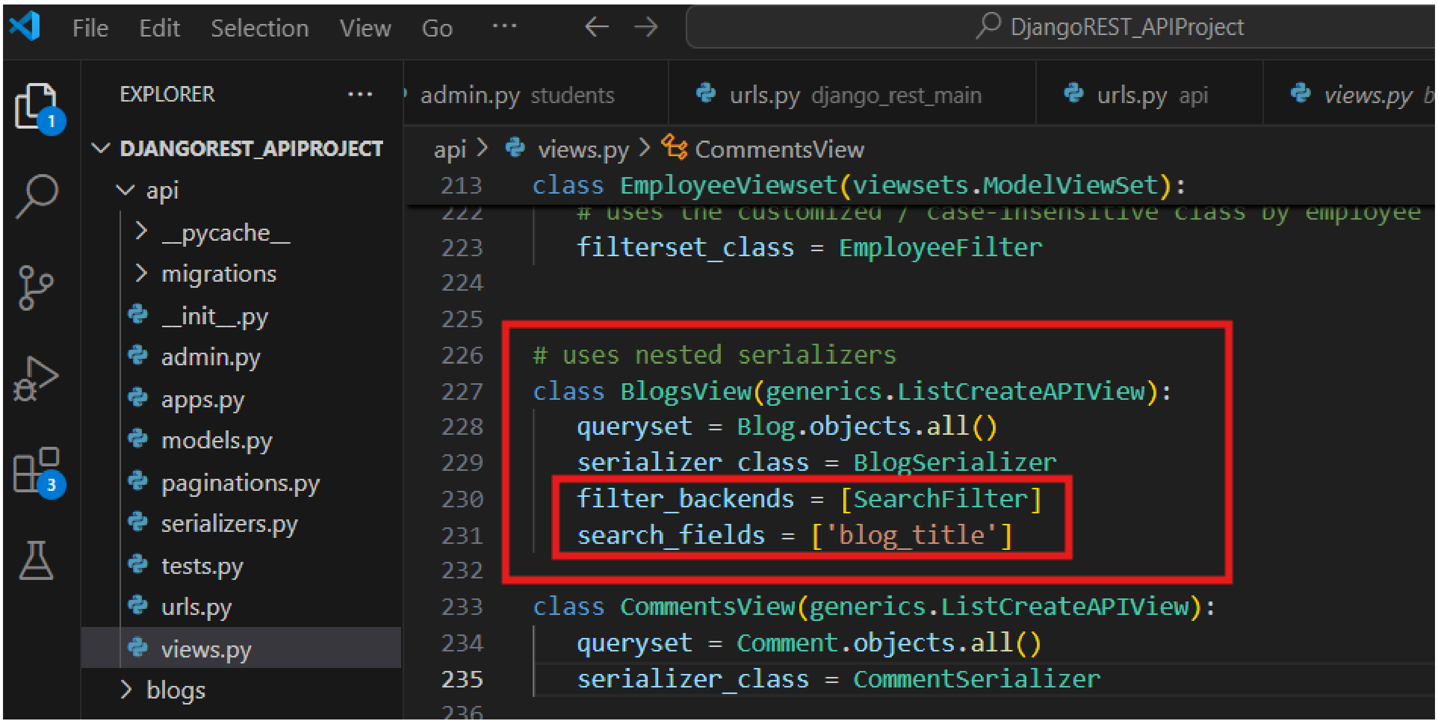
Speaker: Personal | Notebook: API Development using Django Framework

We can search through the database fields like BLOGS' title or comments.

1. Go to API\VIEWS.PY:

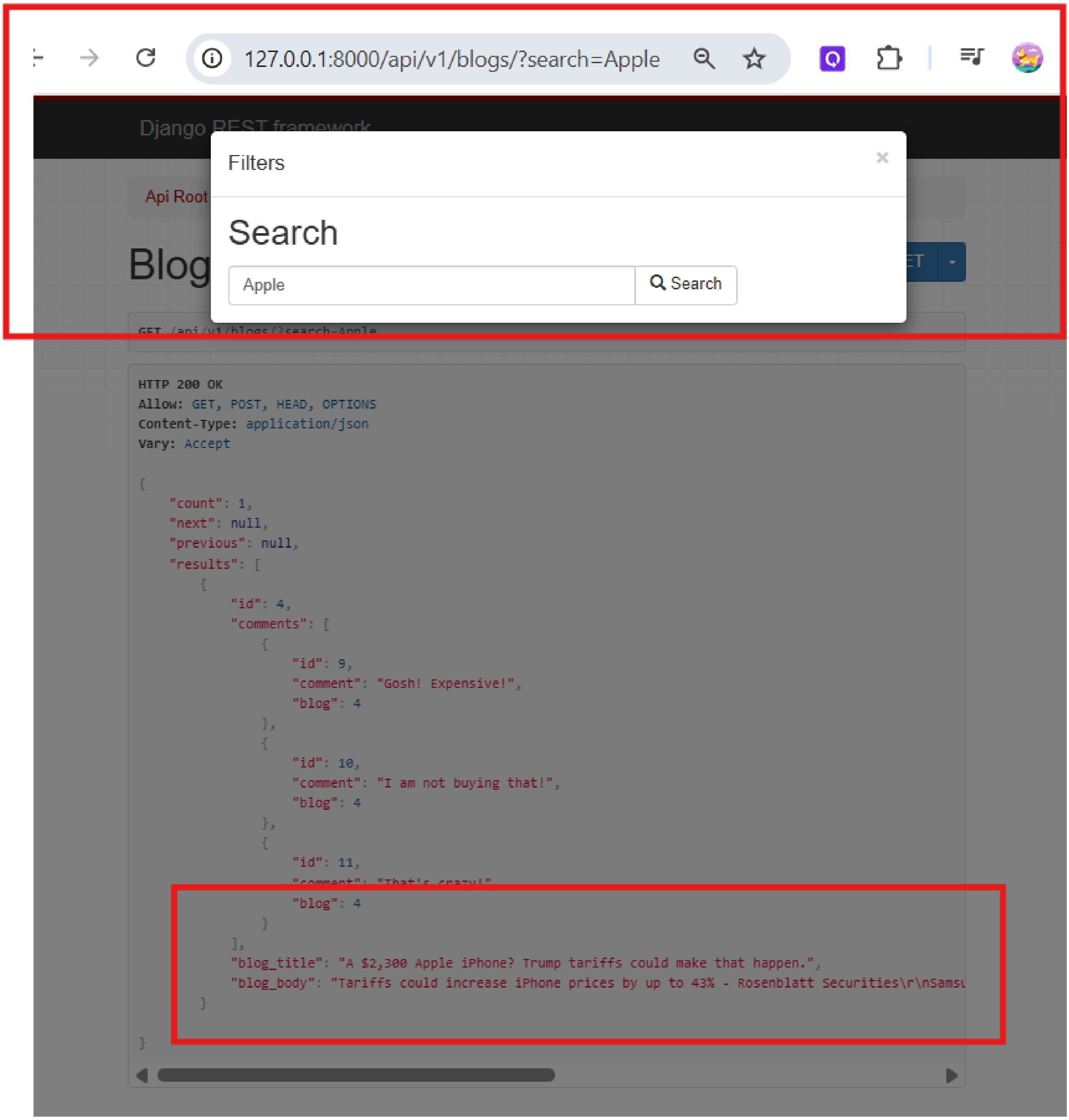


```python
from django.shortcuts import render, get_object_or_404
# from django.http import JsonResponse
from students.models import Student
from employees.models import Employee

from .serializers import StudentSerializer, EmployeeSerializer
from rest_framework.response import Response
from rest_framework import status
from rest_framework.decorators import api_view
from rest_framework.views import APIView
from django.http import Http404

from rest_framework import mixins, generics, viewsets
from blogs.models import Blog, Comment
from blogs.serializers import BlogSerializer, CommentSerializer
from .paginations import CustomPagination
from employees.filters import EmployeeFilter
from rest_framework.filters import SearchFilter
```



```python
class EmployeeViewset(viewsets.ModelViewSet):
    # uses the customized / case-insensitive class by employee
    filterset_class = EmployeeFilter


# uses nested serializers
class BlogsView(generics.ListCreateAPIView):
    queryset = Blog.objects.all()
    serializer_class = BlogSerializer
    filter_backends = [SearchFilter]
    search_fields = ['blog_title']

class CommentsView(generics.ListCreateAPIView):
    queryset = Comment.objects.all()
    serializer_class = CommentSerializer
```

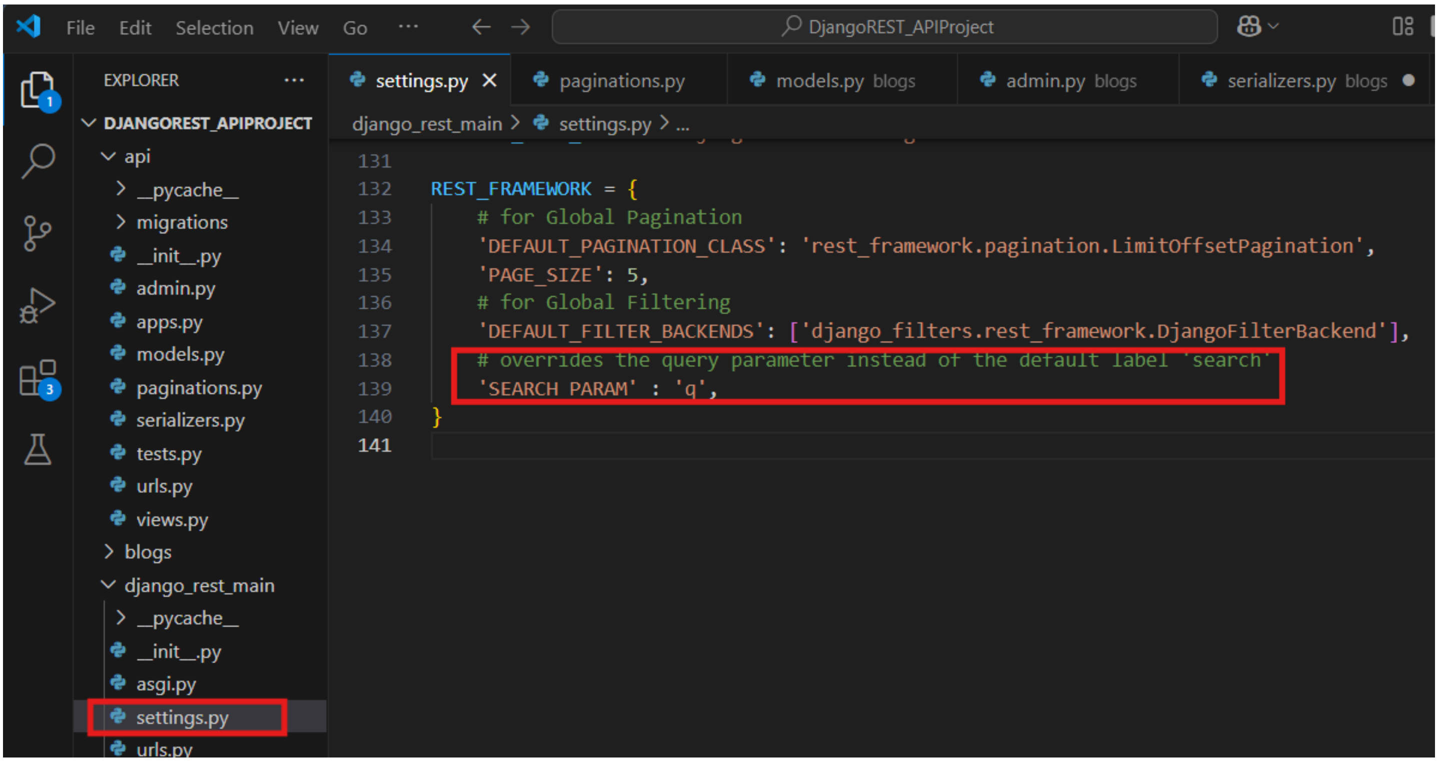
2. Now search thru BLOGS model:



```
GET /api/v1/blogs/?search=Apple
```

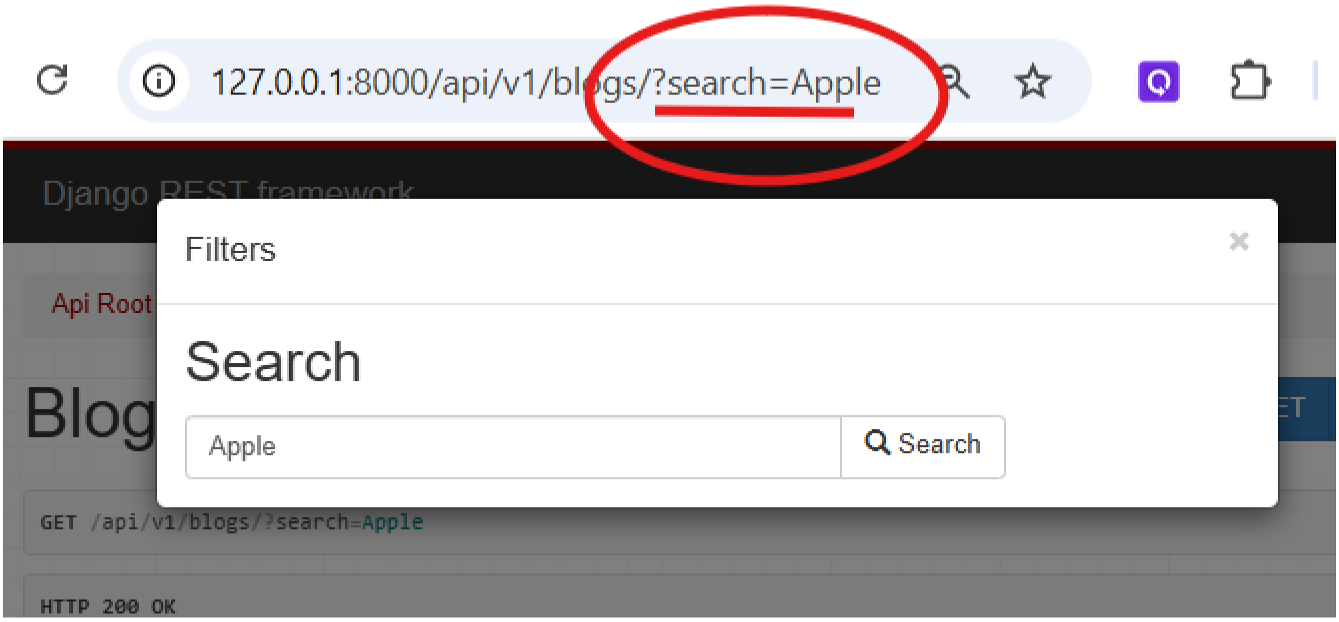
```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 4,
            "comments": [
                {
                    "id": 9,
                    "comment": "Gosh! Expensive!",
                    "blog": 4
                },
                {
                    "id": 10,
                    "comment": "I am not buying that!",
                    "blog": 4
                },
                {
                    "id": 11,
                    "comment": "That's crazy!",
                    "blog": 4
                }
            ],
            "blog_title": "A $2,300 Apple iPhone? Trump tariffs could make that happen.",
            "blog_body": "Tariffs could increase iPhone prices by up to 43% - Rosenblatt Securities\r\nSamsu
        }
    ]
}
```

3. If you want to change the search keyword 'search' to 'q,' then update the SETTINGS.PY



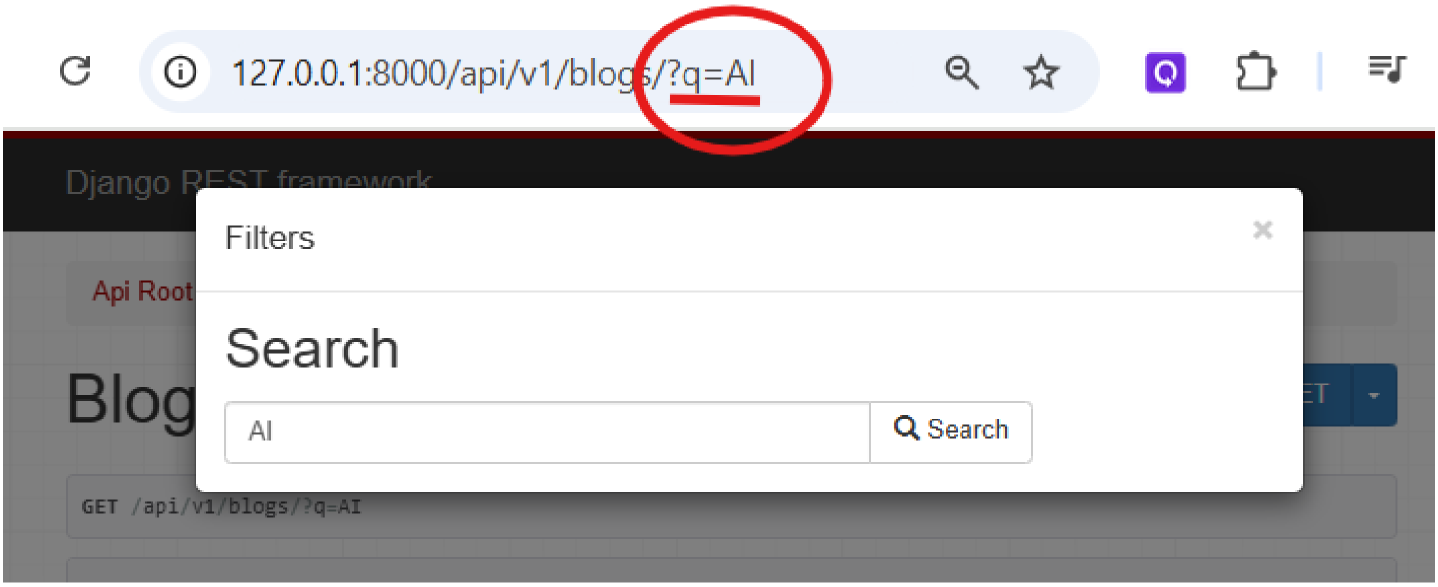
```python
REST_FRAMEWORK = {
    # for Global Pagination
    'DEFAULT_PAGINATION_CLASS': 'rest_framework.pagination.LimitOffsetPagination',
    'PAGE_SIZE': 5,
    # for Global Filtering
    'DEFAULT_FILTER_BACKENDS': ['django_filters.rest_framework.DjangoFilterBackend'],
    # overrides the query parameter instead of the default label 'search'
    'SEARCH_PARAM' : 'q',
}
```

FROM:



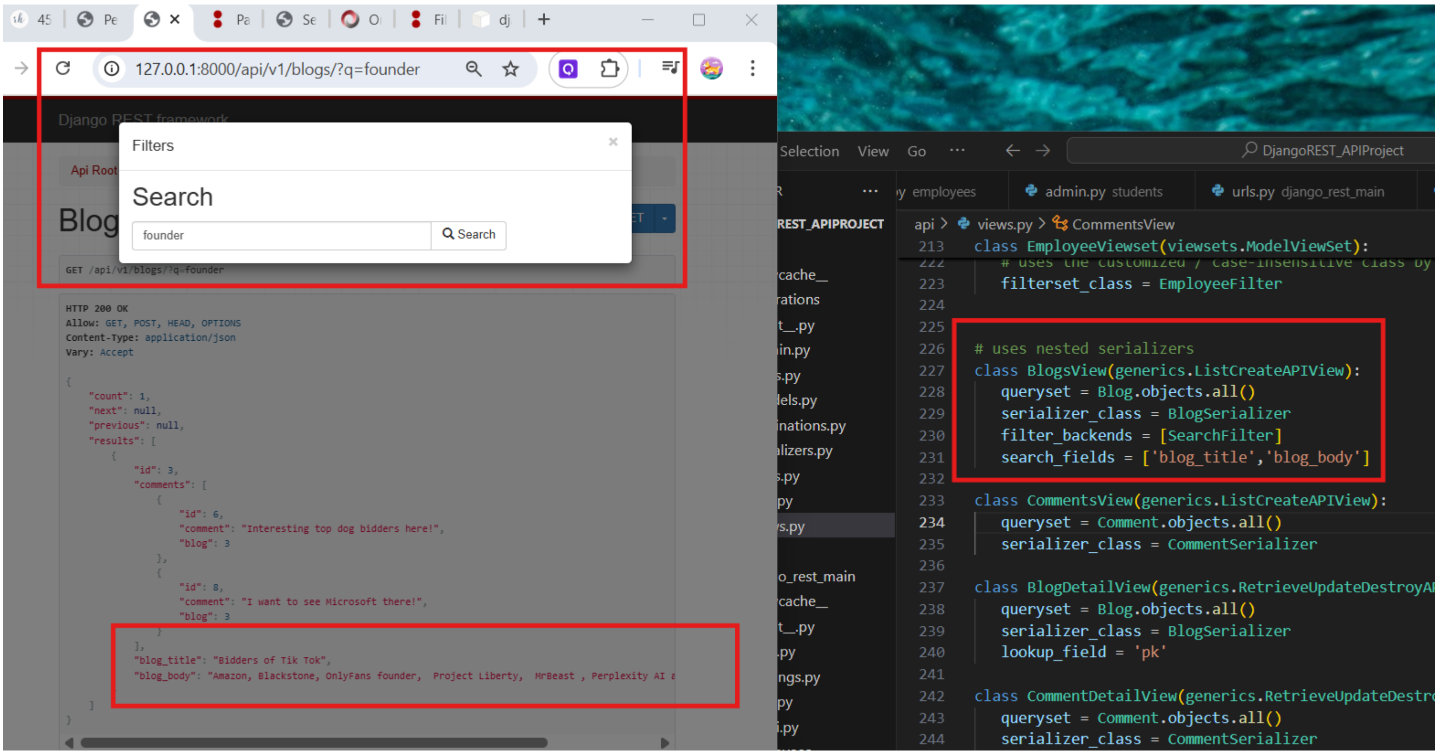
```
GET /api/v1/blogs/?search=Apple

HTTP 200 OK
```

TO:



```
GET /api/v1/blogs/?q=AI
```

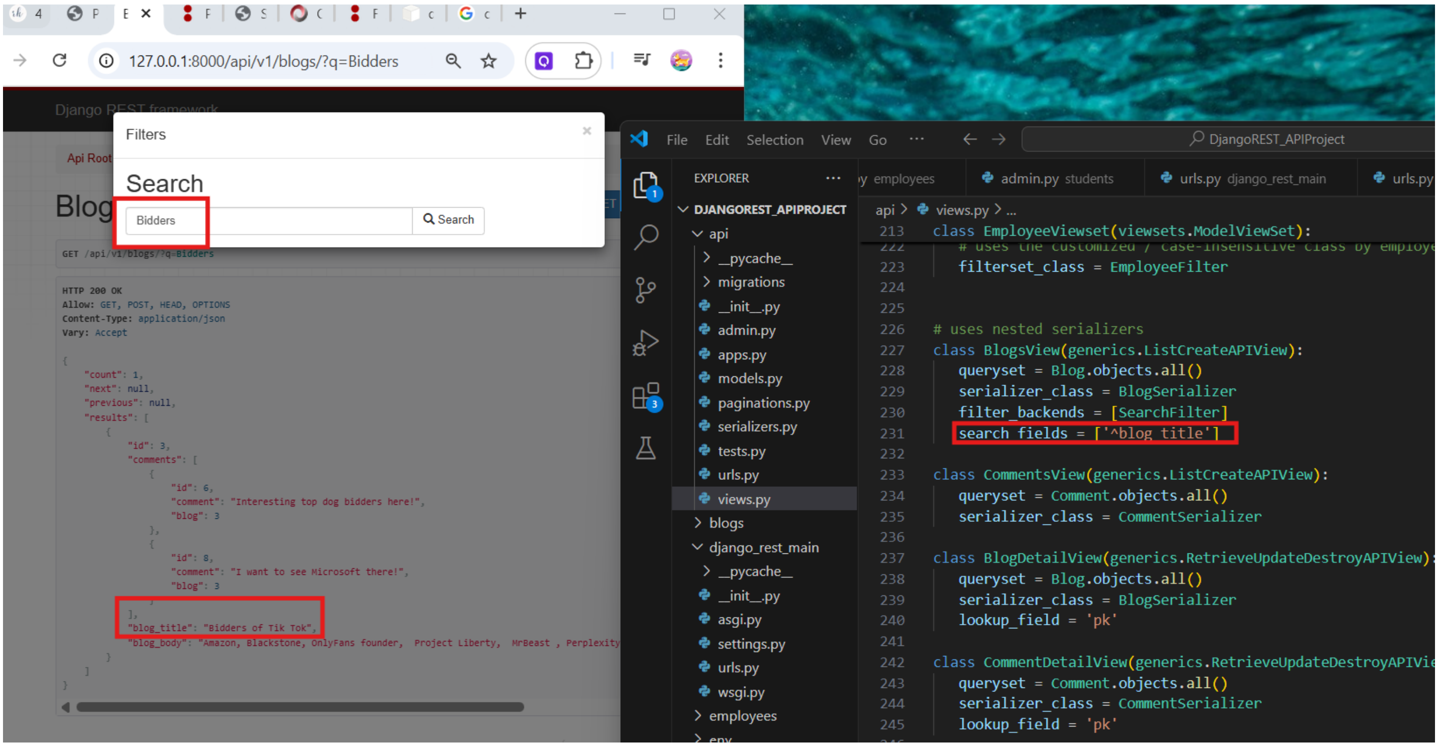
4. To search through the BLOG_BODY:



```
GET /api/v1/blogs/?q=founder

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

{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 3,
            "comments": [
                {
                    "id": 6,
                    "comment": "Interesting top dog bidders here!",
                    "blog": 3
                },
                {
                    "id": 8,
                    "comment": "I want to see Microsoft there!",
                    "blog": 3
                }
            ],
            "blog_title": "Bidders of Tik Tok",
            "blog_body": "Amazon, Blackstone, OnlyFans founder,  Project Liberty,  MrBeast , Perplexity AI a
        }
    ]
}
```

```python
class EmployeeViewset(viewsets.ModelViewSet):
    # uses the customized / case-insensitive class by
    filterset_class = EmployeeFilter


# uses nested serializers
class BlogsView(generics.ListCreateAPIView):
    queryset = Blog.objects.all()
    serializer_class = BlogSerializer
    filter_backends = [SearchFilter]
    search_fields = ['blog_title','blog_body']

class CommentsView(generics.ListCreateAPIView):
    queryset = Comment.objects.all()
    serializer_class = CommentSerializer

class BlogDetailView(generics.RetrieveUpdateDestroyAP
    queryset = Blog.objects.all()
    serializer_class = BlogSerializer
    lookup_field = 'pk'

class CommentDetailView(generics.RetrieveUpdateDestro
    queryset = Comment.objects.all()
    serializer_class = CommentSerializer
```

5. To search for a blog title that starts with a specific keyword, use the caret symbol (^).

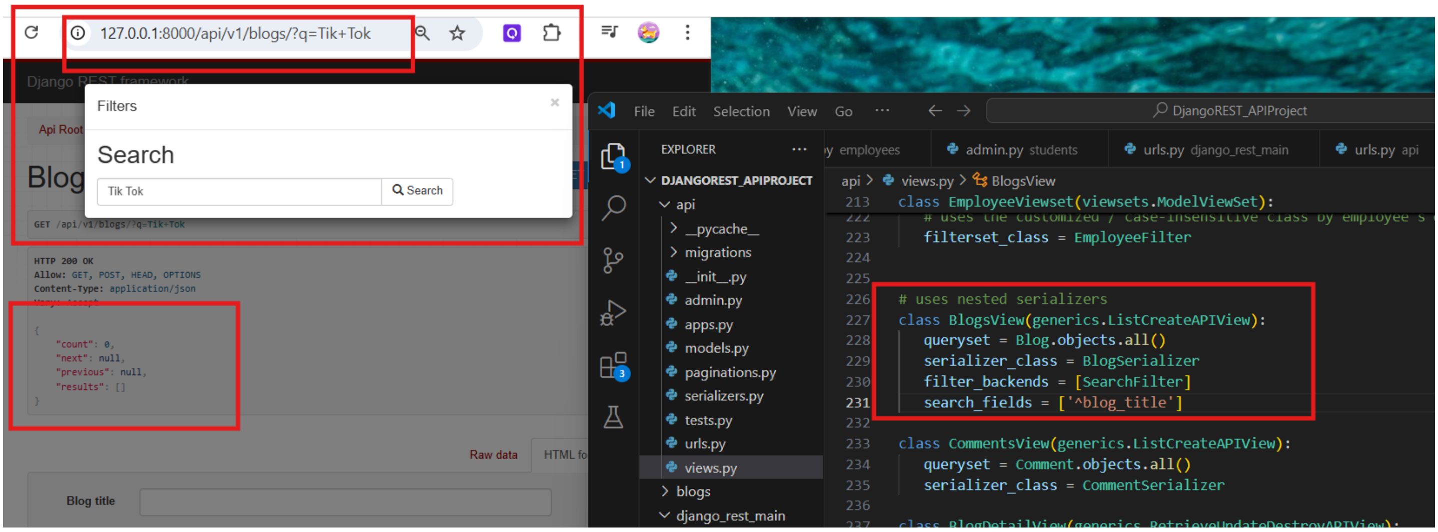


127.0.0.1:8000/api/v1/blogs/?q=Bidders

```
GET /api/v1/blogs/?q=Bidders

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

{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 3,
            "comments": [
                {
                    "id": 6,
                    "comment": "Interesting top dog bidders here!",
                    "blog": 3
                },
                {
                    "id": 8,
                    "comment": "I want to see Microsoft there!",
                    "blog": 3
                }
            ],
            "blog_title": "Bidders of Tik Tok",
            "blog_body": "Amazon, Blackstone, OnlyFans founder,  Project Liberty,  MrBeast , Perplexity
        }
    ]
}
```

```python
class EmployeeViewset(viewsets.ModelViewSet):
    # uses the customized / case-insensitive class by employe
    filterset_class = EmployeeFilter


# uses nested serializers
class BlogsView(generics.ListCreateAPIView):
    queryset = Blog.objects.all()
    serializer_class = BlogSerializer
    filter_backends = [SearchFilter]
    search_fields = ['^blog_title']

class CommentsView(generics.ListCreateAPIView):
    queryset = Comment.objects.all()
    serializer_class = CommentSerializer

class BlogDetailView(generics.RetrieveUpdateDestroyAPIView):
    queryset = Blog.objects.all()
    serializer_class = BlogSerializer
    lookup_field = 'pk'

class CommentDetailView(generics.RetrieveUpdateDestroyAPIVie
    queryset = Comment.objects.all()
    serializer_class = CommentSerializer
    lookup_field = 'pk'
```

But when you use a keyword, not the actual starting word, it won't show any result.



127.0.0.1:8000/api/v1/blogs/?q=Tik+Tok

```
GET /api/v1/blogs/?q=Tik+Tok

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

{
    "count": 0,
    "next": null,
    "previous": null,
    "results": []
}
```

```python
class EmployeeViewset(viewsets.ModelViewSet):
    # uses the customized / case-insensitive class by employee s
    filterset_class = EmployeeFilter


# uses nested serializers
class BlogsView(generics.ListCreateAPIView):
    queryset = Blog.objects.all()
    serializer_class = BlogSerializer
    filter_backends = [SearchFilter]
    search_fields = ['^blog_title']

class CommentsView(generics.ListCreateAPIView):
    queryset = Comment.objects.all()
    serializer_class = CommentSerializer

class BlogDetailView(generics.RetrieveUpdateDestroyAPIView):
```