

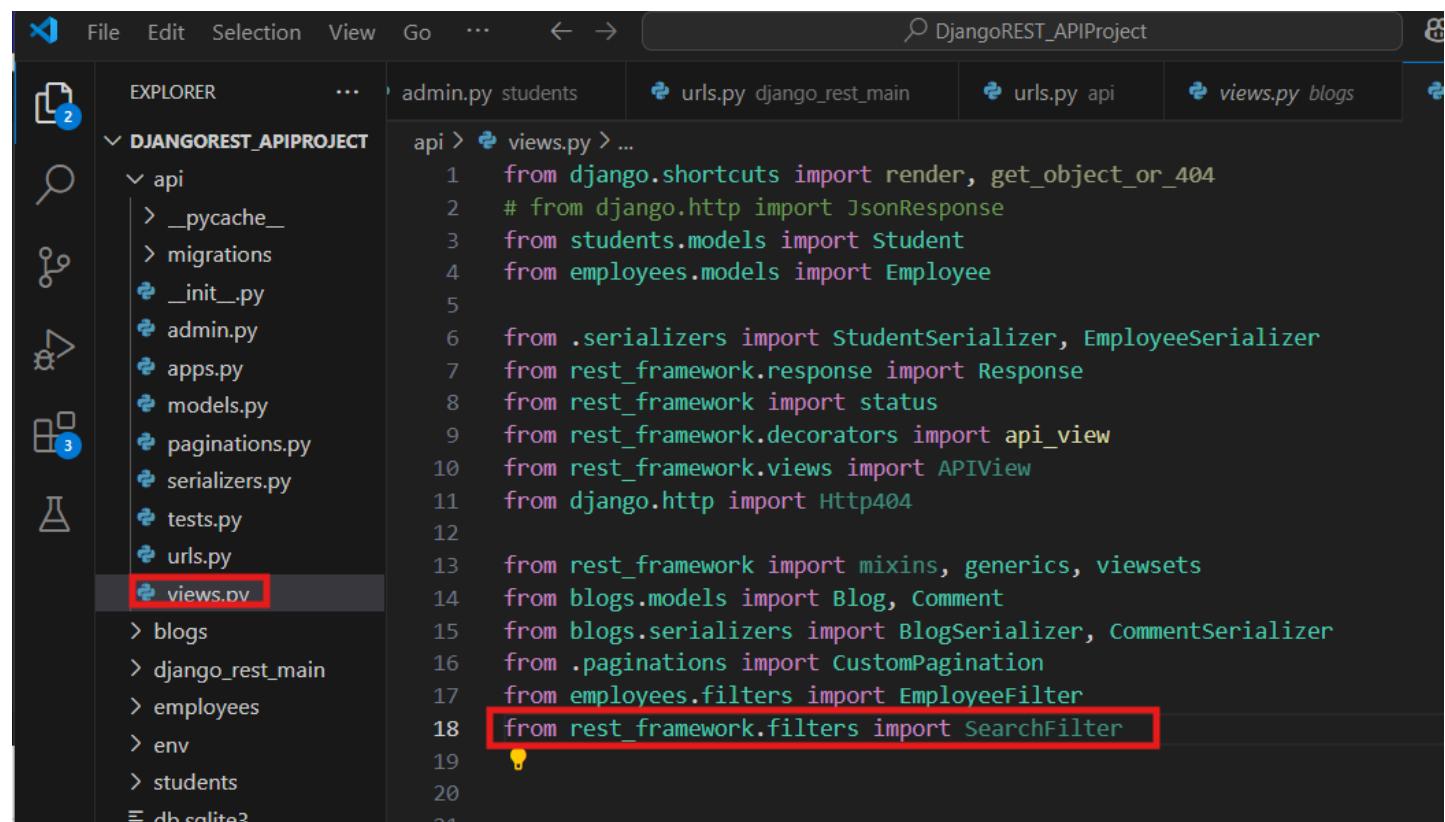
## Topic: 16. DRF Searching

Speaker: Personal / Notebook: API Development using Django Framework



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

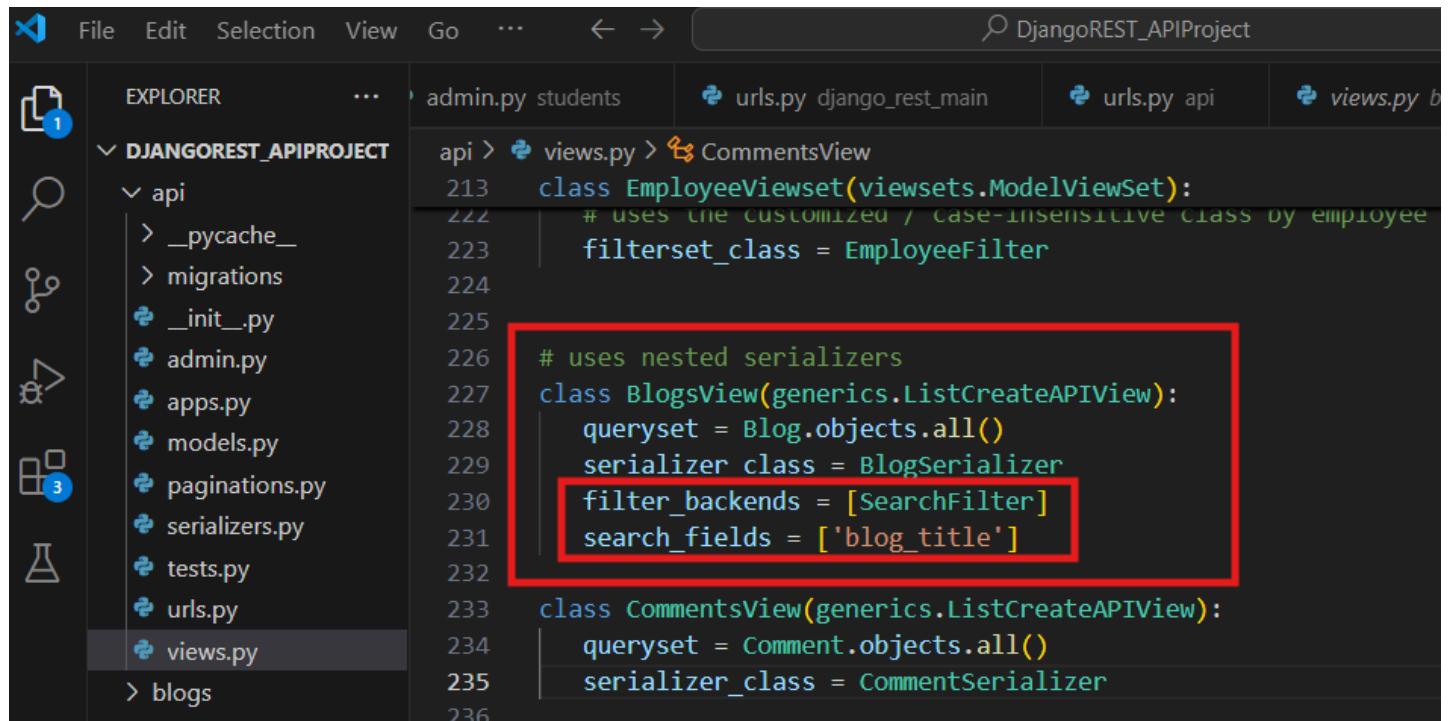
1. Go to API\VIEWS.PY:



The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with icons for file operations, a search bar, and a list of project files under 'DJANGOREST\_APIPROJECT'. The 'views.py' file is selected and highlighted with a red box around line 18. The code editor shows the following Python code:

```
1  from django.shortcuts import render, get_object_or_404
2  # from django.http import JsonResponse
3  from students.models import Student
4  from employees.models import Employee
5
6  from .serializers import StudentSerializer, EmployeeSerializer
7  from rest_framework.response import Response
8  from rest_framework import status
9  from rest_framework.decorators import api_view
10 from rest_framework.views import APIView
11 from django.http import Http404
12
13 from rest_framework import mixins, generics, viewsets
14 from blogs.models import Blog, Comment
15 from blogs.serializers import BlogSerializer, CommentSerializer
16 from .paginations import CustomPagination
17 from employees.filters import EmployeeFilter
18 from rest_framework.filters import SearchFilter
```

The line 'from rest\_framework.filters import SearchFilter' is highlighted with a red box.



File Edit Selection View Go ... ← → 🔍 DjangoREST\_APIProject

EXPLORER ...

DJANGOREST\_APIPROJECT

- api
  - \_\_pycache\_\_
  - migrations
  - \_\_init\_\_.py
  - admin.py
  - apps.py
  - models.py
  - paginations.py
  - serializers.py
  - tests.py
  - urls.py
  - views.py
- blogs

```
admin.py students urls.py django_rest_main urls.py api views.py b
api > views.py > CommentsView
213 class EmployeeViewSet(viewsets.ModelViewSet):
222     # uses the customized / case-insensitive class by employee
223     filterset_class = EmployeeFilter
224
225
226     # uses nested serializers
227 class BlogsView(generics.ListCreateAPIView):
228     queryset = Blog.objects.all()
229     serializer_class = BlogSerializer
230     filter_backends = [SearchFilter]
231     search_fields = ['blog_title']
232
233 class CommentsView(generics.ListCreateAPIView):
234     queryset = Comment.objects.all()
235     serializer_class = CommentSerializer
236
```

2. Now search thru BLOGS model:

The screenshot shows a browser window with the URL `127.0.0.1:8000/api/v1/blogs/?search=Apple`. A red box highlights the search bar and the results table. A second red box highlights the 'blog' field in the JSON response.

**Filters**

**Search**

Apple

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

HTTP 200 OK

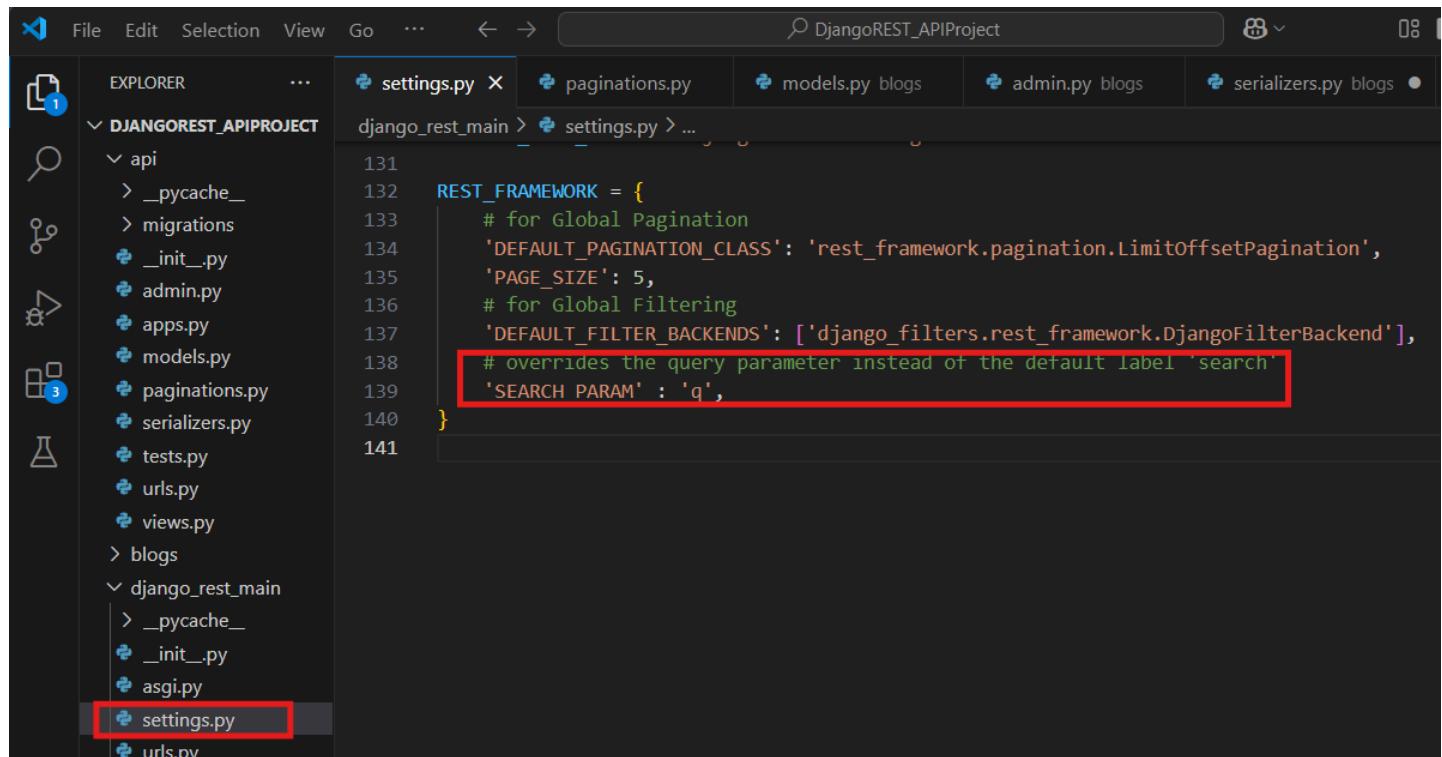
Allow: GET, POST, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

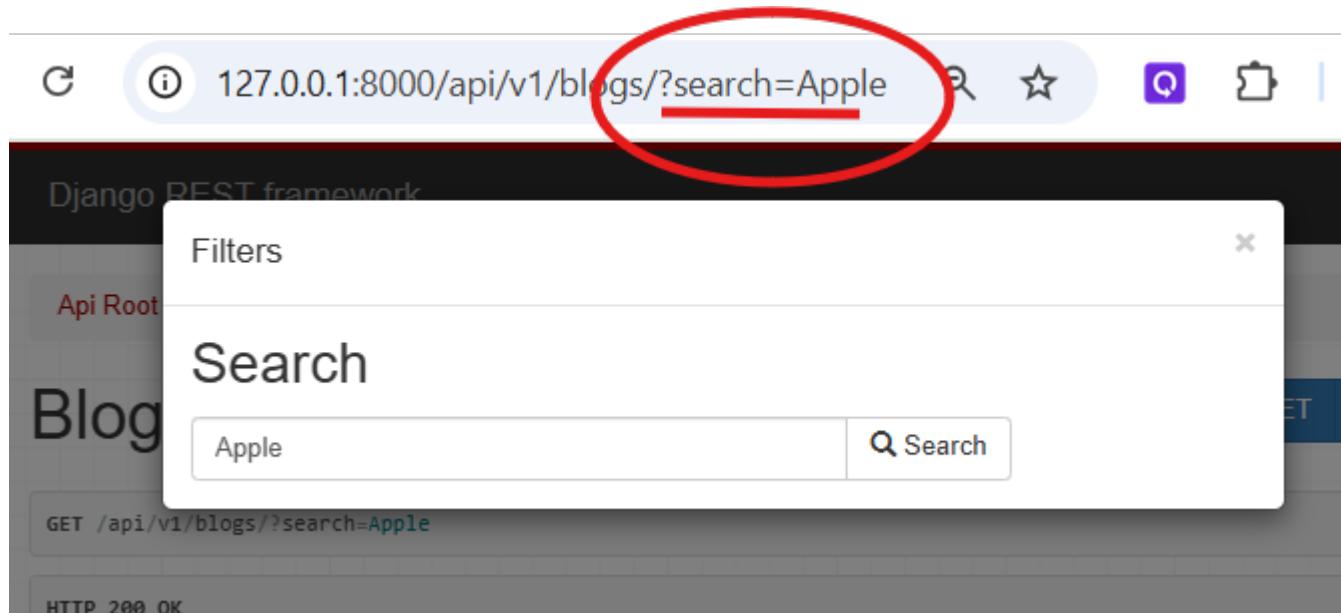
```
{ "count": 1, "next": null, "previous": null, "results": [ { "id": 4, "comments": [ { "id": 9, "comment": "Gosh! Expensive!", "blog": 4 }, { "id": 10, "comment": "I am not buying that!", "blog": 4 }, { "id": 11, "comment": "That's crazy!", "blog": 4 } ], "blog_title": "A $2,300 Apple iPhone? Trump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

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



```
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:



127.0.0.1:8000/api/v1/blogs/?search=Apple

Django REST framework

Filters

Api Root

Blog

Search

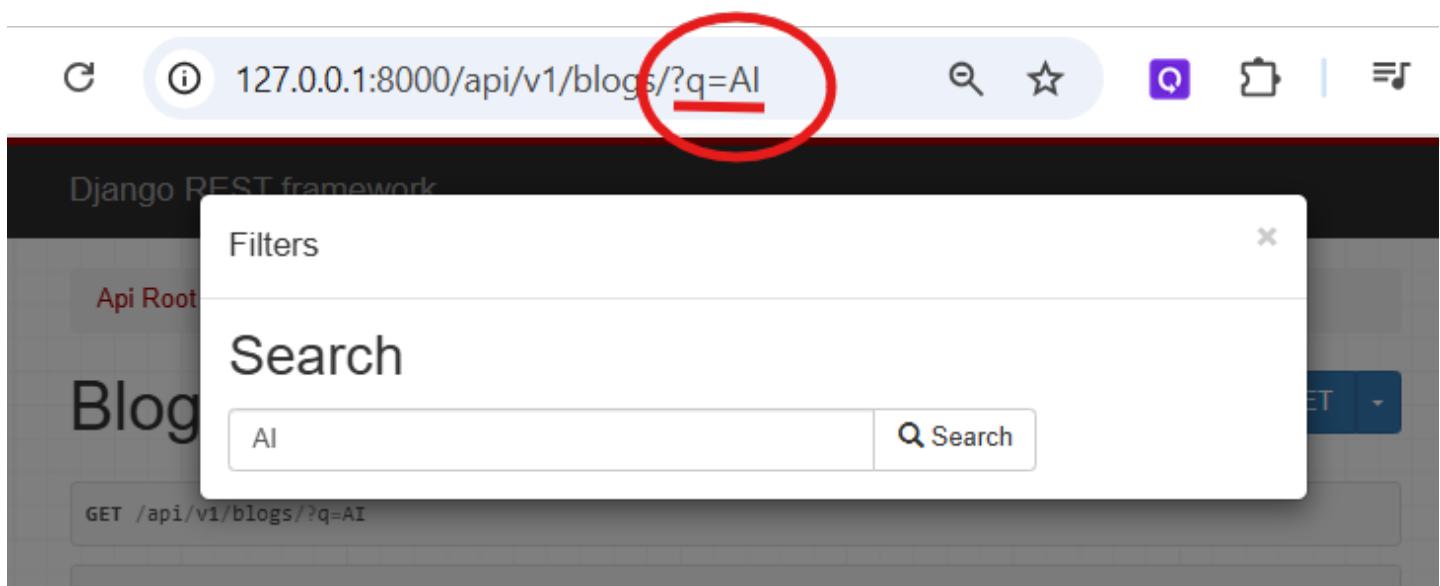
Apple

Search

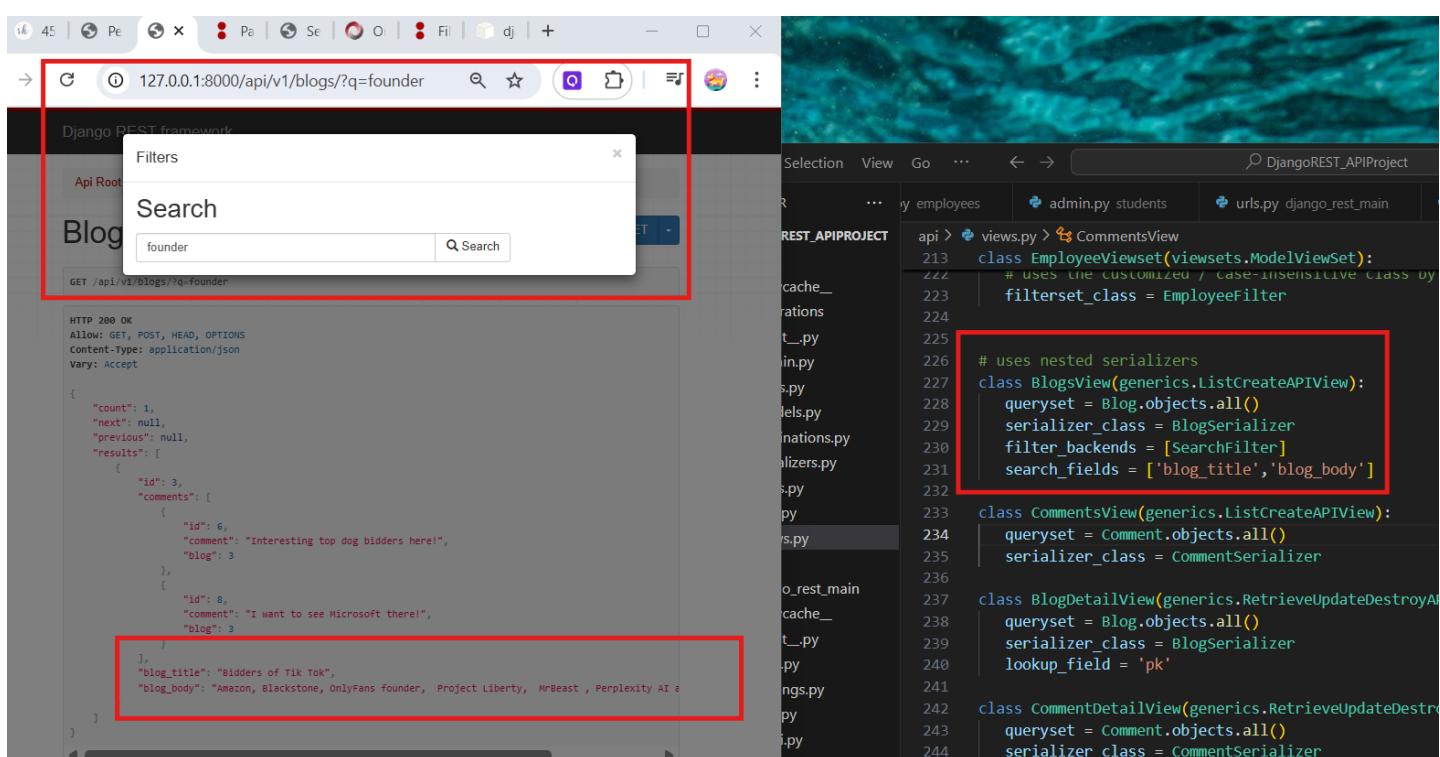
GET /api/v1/blogs/?search=Apple

HTTP 200 OK

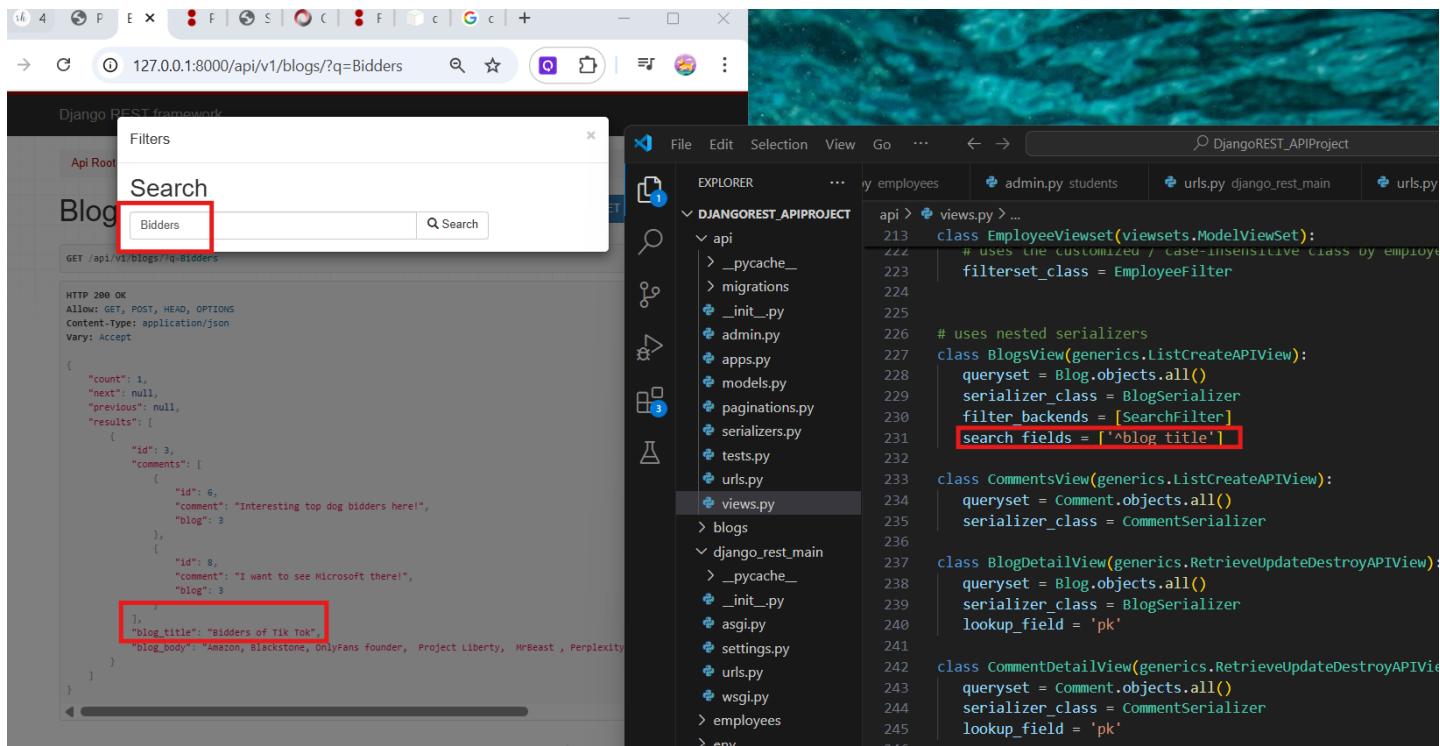
TO:



4. To search through the BLOG\_BODY:



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

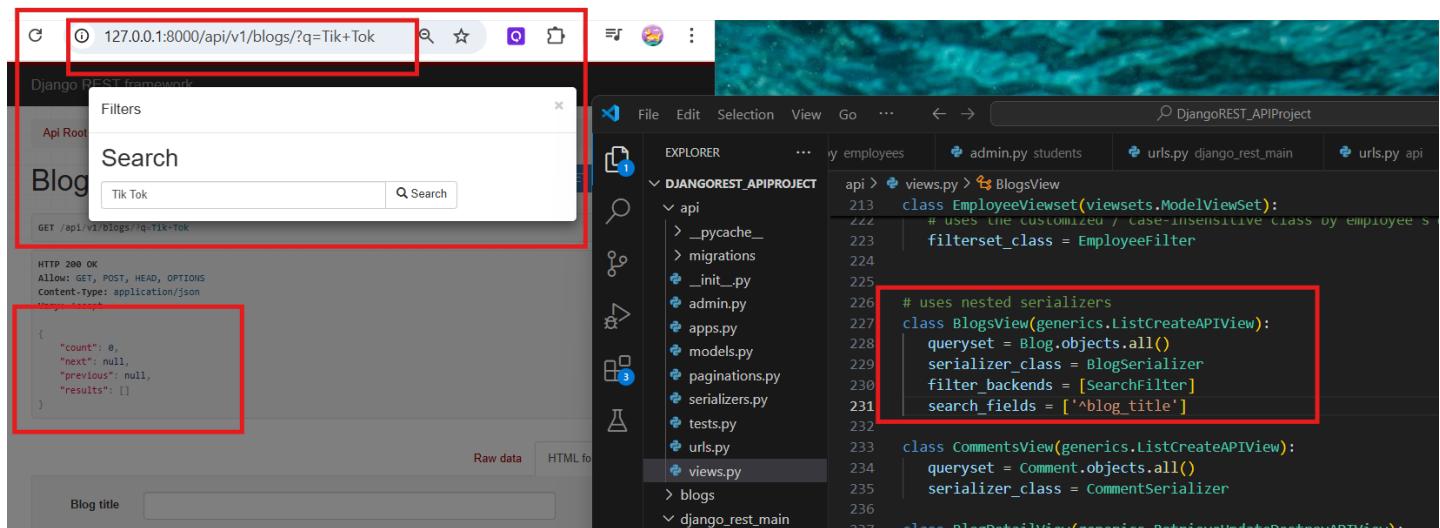


The screenshot shows a browser window with the URL `127.0.0.1:8000/api/v1/blogs/?q=Bidders`. The response is a JSON object with a count of 1, containing a single blog entry with an ID of 3, a comment, and a blog body. The code editor on the right shows the `views.py` file for the `api` application, specifically the `BlogsView` class. The `search_fields` attribute is set to `['^blog_title']`, which matches the search term 'Bidders'.

```

213     class BlogsView(generics.ListCreateAPIView):
214         queryset = Blog.objects.all()
215         serializer_class = BlogSerializer
216         filter_backends = [SearchFilter]
217         search_fields = ['^blog_title']
  
```

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



The screenshot shows a browser window with the URL `127.0.0.1:8000/api/v1/blogs/?q=Tik+Tok`. The response is a JSON object with a count of 0, indicating no results found. The code editor on the right shows the `views.py` file for the `api` application, specifically the `BlogsView` class. The `search_fields` attribute is set to `['^blog_title']`, which does not match the search term 'Tik Tok'.

```

213     class BlogsView(generics.ListCreateAPIView):
214         queryset = Blog.objects.all()
215         serializer_class = BlogSerializer
216         filter_backends = [SearchFilter]
217         search_fields = ['^blog_title']
  
```