

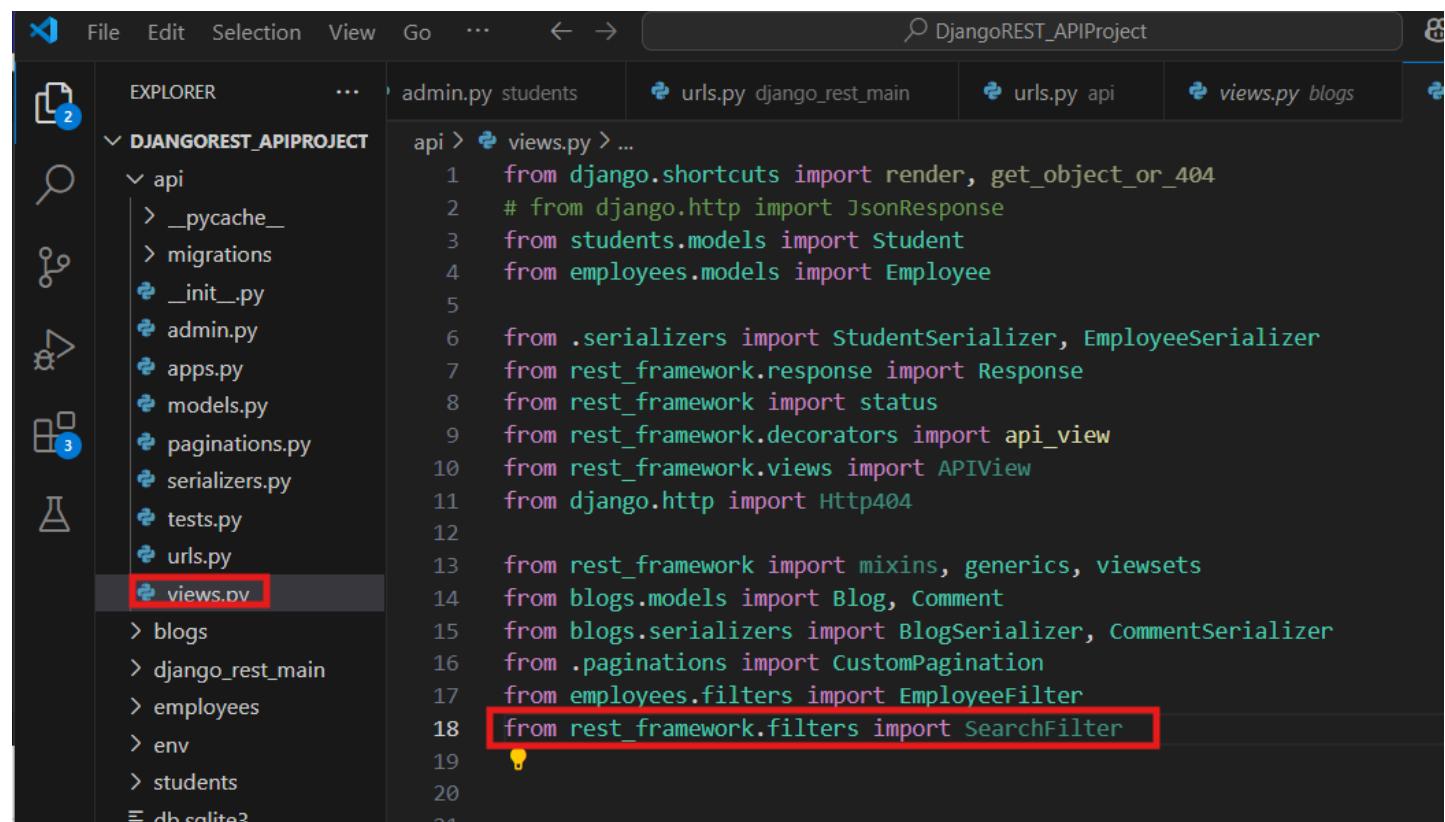
Topic: 16. DRF Searching

Speaker: Personal / Notebook: API Development using Django Framework



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

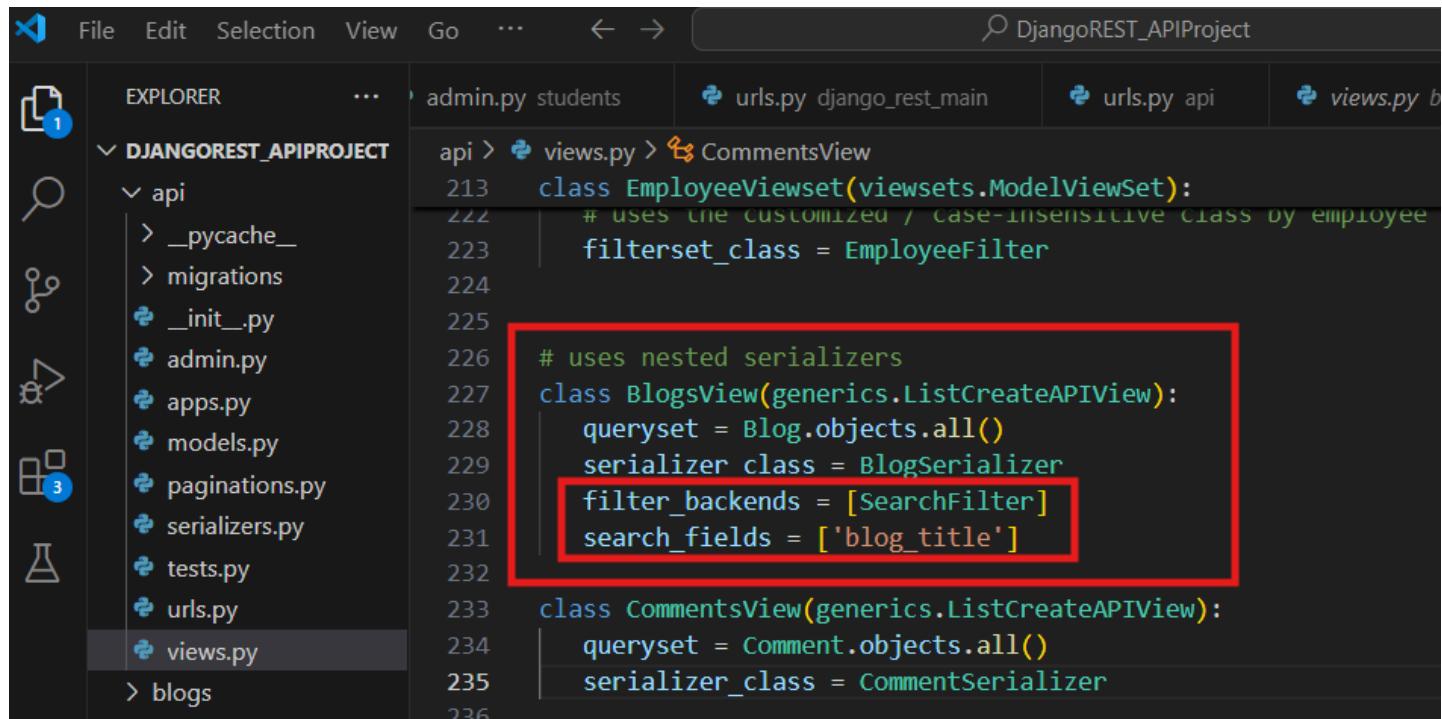
1. Go to API\VIEWS.PY:



The screenshot shows a code editor with a dark theme. The left sidebar is an 'EXPLORER' view showing the project structure of 'DJANGOREST_APIPROJECT'. The 'api' folder is expanded, showing 'views.py' as the active file. 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. The code editor has a search bar at the top right with the text 'DjangoREST_APIProject'.



File Edit Selection View Go ... ← → 🔍 DjangoREST_APIProject

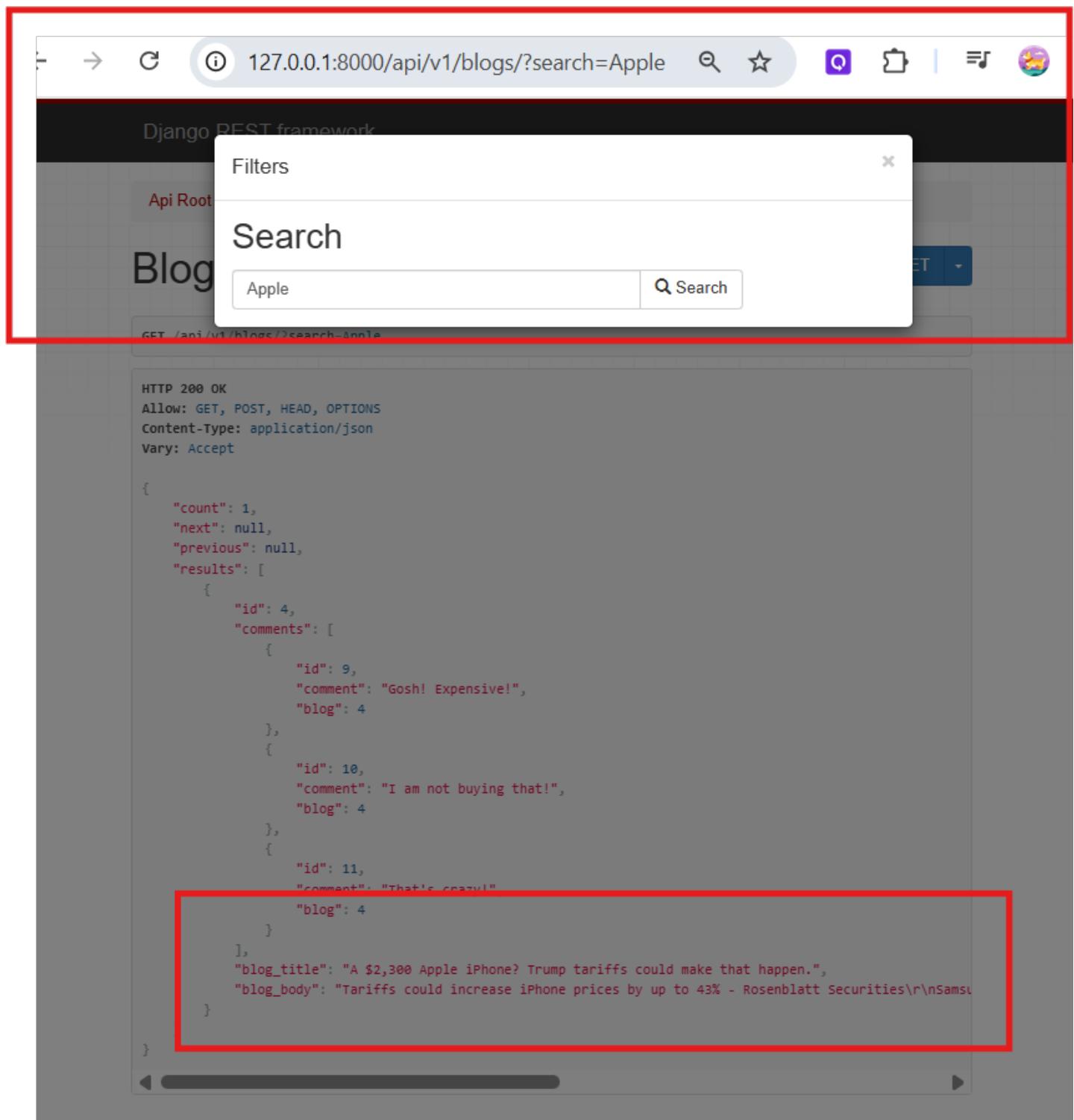
EXPLORER ...

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:

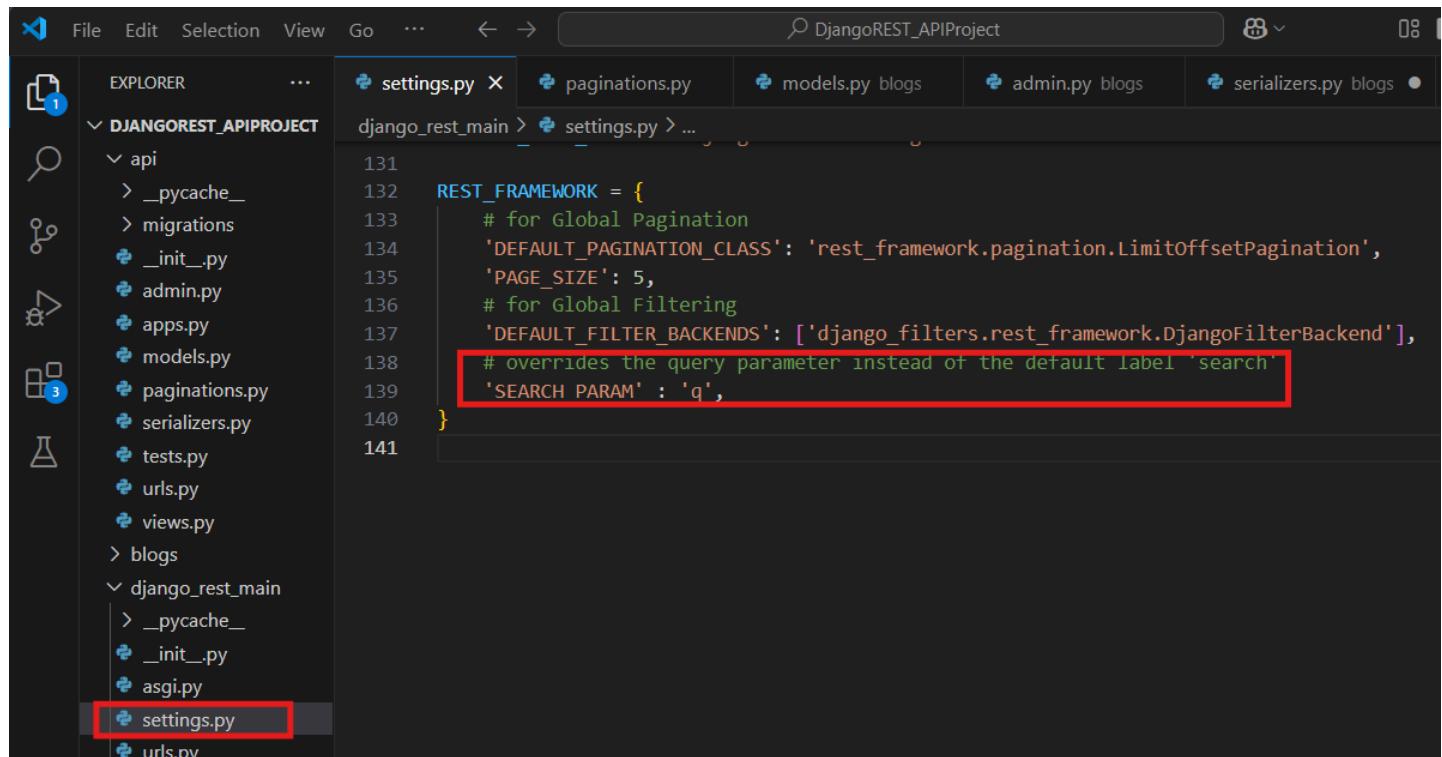


The screenshot shows a browser window with the URL `127.0.0.1:8000/api/v1/blogs/?search=Apple`. A modal dialog titled "Search" is open, containing a search input field with the value "Apple" and a "Search" button. Below the modal, the API response is displayed in a code block:

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

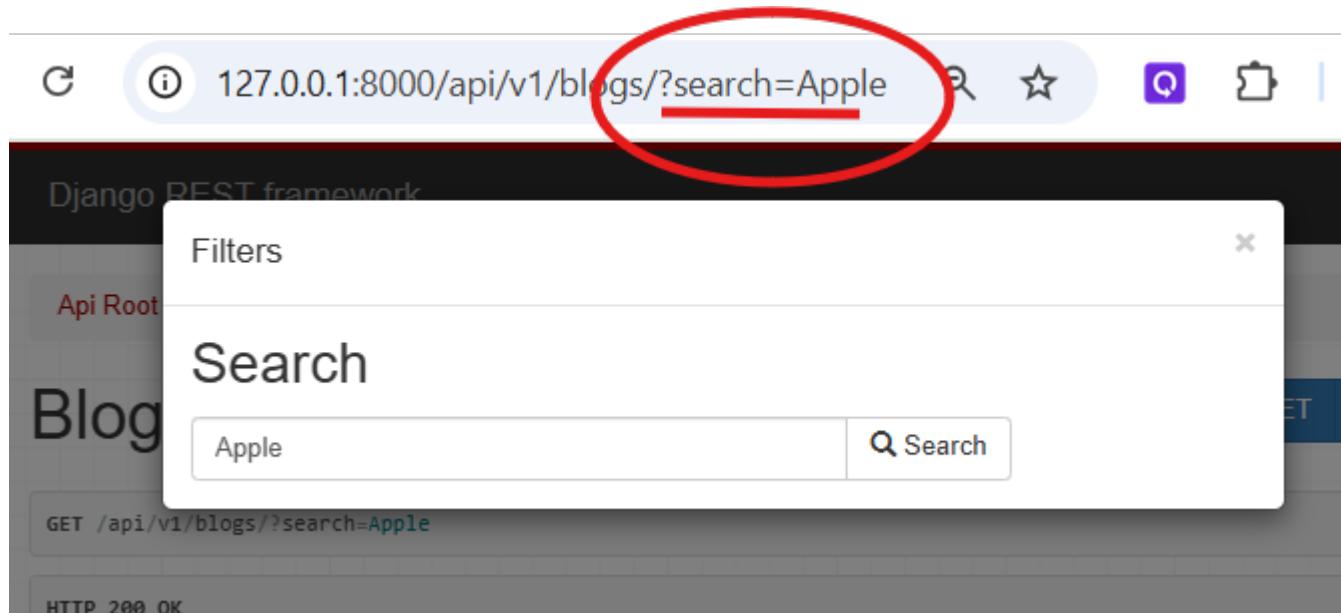
{
    "count": 1,
    "next": null,
    "previous": null,
    "results": [
        {
            "id": 4,
            "comments": [
                {
                    "id": 9,
                    "comment": "Gosh! Expensive!",
                    "blog": 4
                },
                {
                    "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

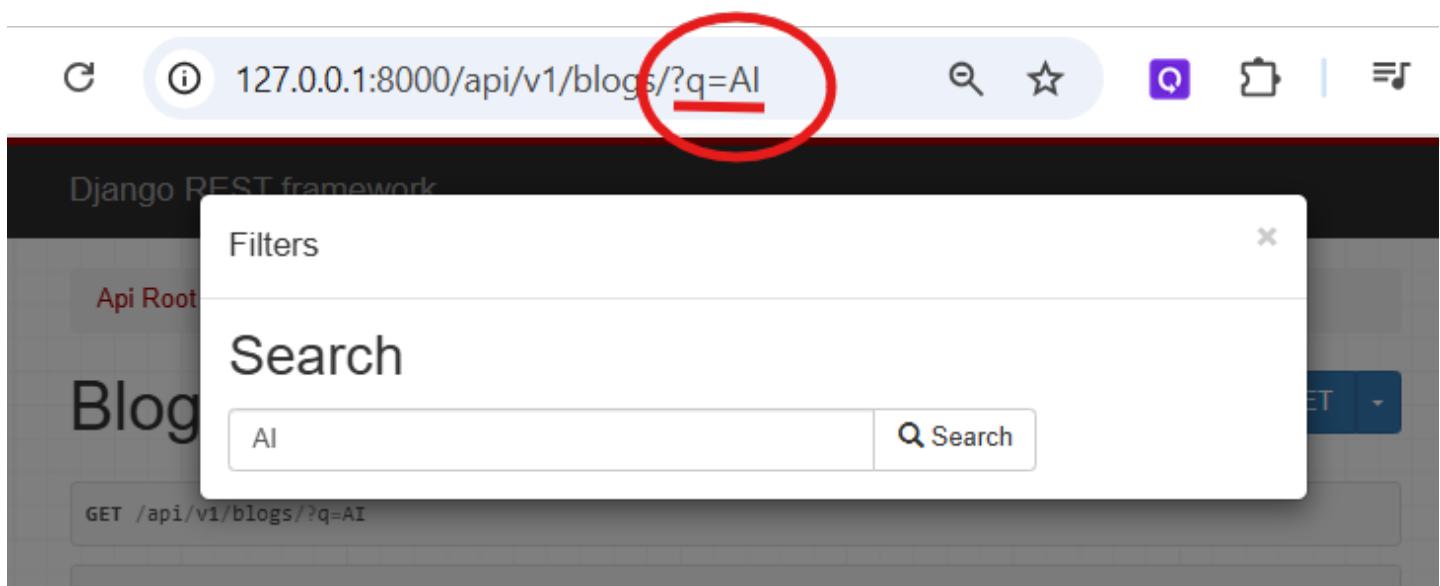
Apple

Search

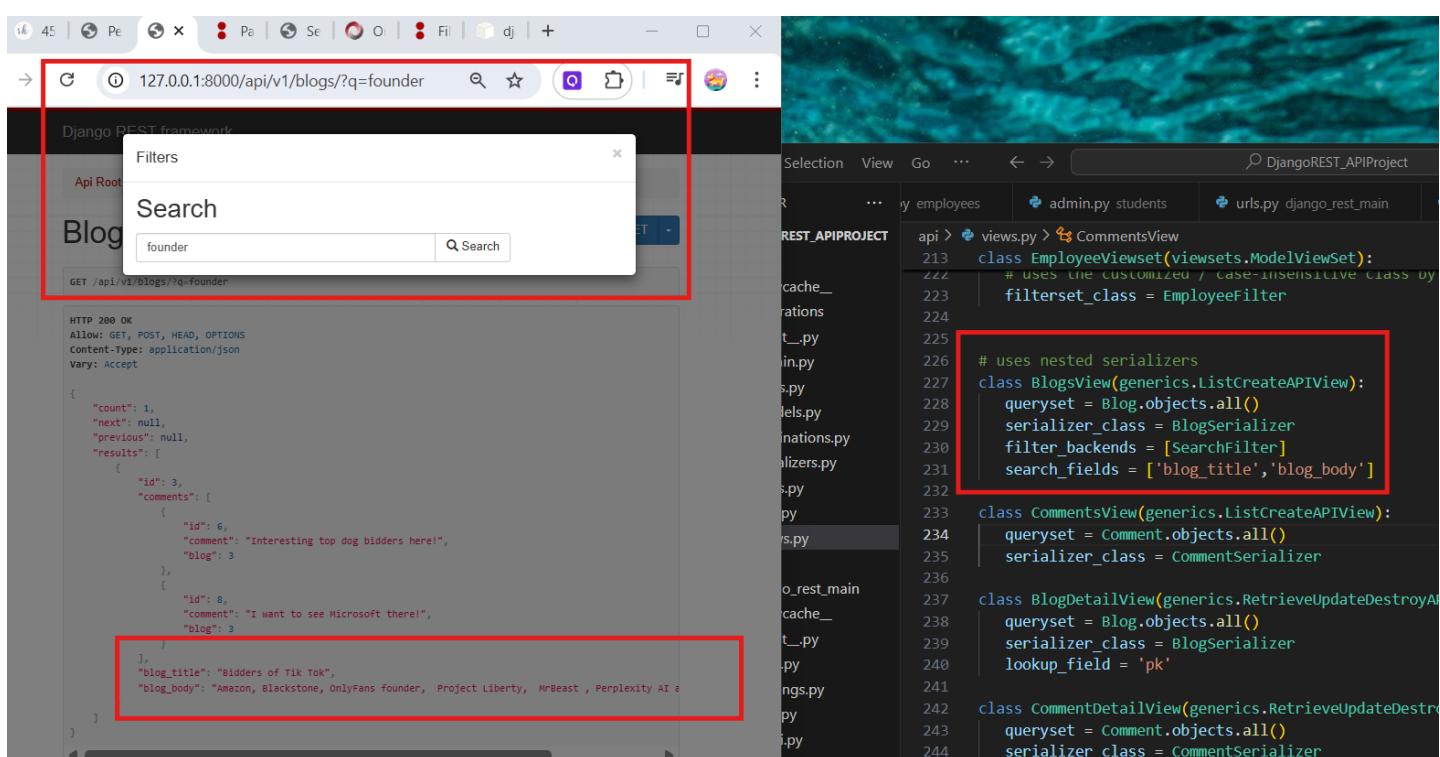
GET /api/v1/blogs/?search=Apple

HTTP 200 OK

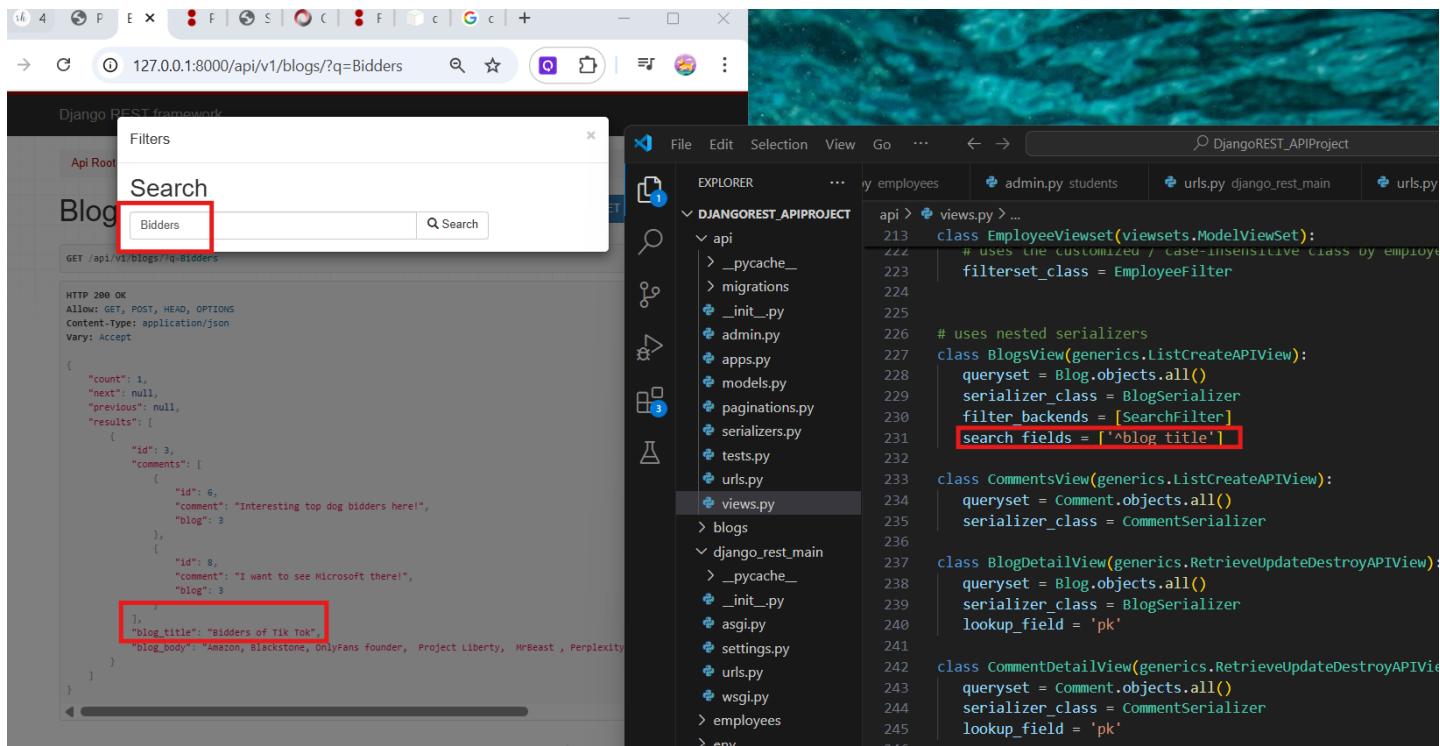
TO:



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, 'next' and 'previous' null, and a single result object. The result has an 'id' of 3, a 'comment' of 'Interesting top dog bidders here!', and a 'blog' of 3. The 'blog' object has an 'id' of 8, a 'comment' of 'I want to see Microsoft there!', and a 'blog' of 3. The 'blog' object also has a 'blog_title' of 'Bidders of Tik Tok' and a 'blog_body' of 'Amazon, Blackstone, OnlyFans founder, Project Liberty, MrBeast, Perplexity'. A red box highlights the 'Bidders' search term in the browser's search bar and the 'blog_title' field in the JSON response.

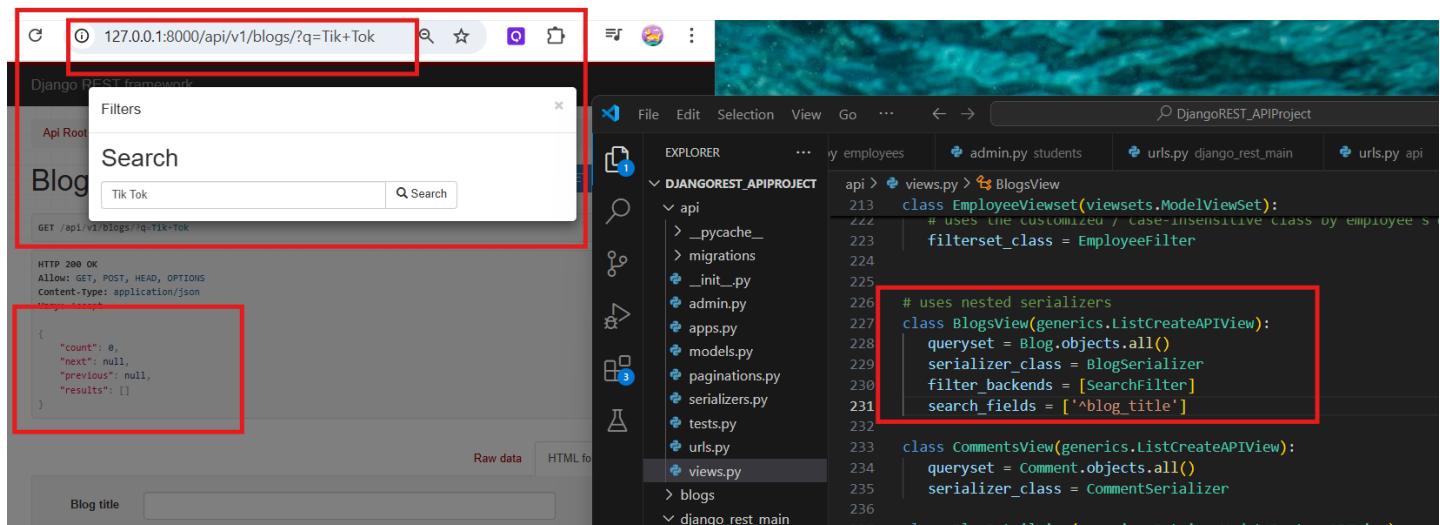
The code editor on the right shows the `views.py` file from the `DJANGOREST_APIPROJECT` project. The `search_fields` attribute is highlighted with a red box, containing the value `['^blog_title']`.

```

class BlogsView(generics.ListCreateAPIView):
    queryset = Blog.objects.all()
    serializer_class = BlogSerializer
    filter_backends = [SearchFilter]
    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, 'next' and 'previous' null, and an empty 'results' array. A red box highlights the 'Tik Tok' search term in the browser's search bar and the empty 'results' array in the JSON response.

The code editor on the right shows the `views.py` file from the `DJANGOREST_APIPROJECT` project. The `search_fields` attribute is highlighted with a red box, containing the value `['^blog_title']`.

```

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

```