



리액트와 함께 장고 시작하기 / 장고 DRF

# Filtering과 Ordering

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

# Filtering

목록조회 APIView에서는 조건에 따라 필터링이 필요  
(ex: QuerySet의 filter/exclude 등)

APIView 역시 Class Based View

필터링에 필요한 인자 참조

APIView	@api_view
self.request.user	request.user
self.request.GET	request.GET
self.request.query_params (GET과 동일)	request.query_params (GET과 동일)
self.kwargs	함수의 키워드 인자

# 직접 Filtering의 예 → ListAPIView에서는 get\_queryset 재정의

```
from rest_framework import generics

class PostListAPIView(generics.ListAPIView):
    queryset = Post.objects.all()

    def get_queryset(self):
        q = self.request.query_params.get('q', '')
        qs = super().get_queryset()
        if q:
            qs = qs.filter(title__icontains=q) # 적절히 필터링
        return qs
```

# Generic Filtering / Ordering

## Django Admin의 search 기능과 유사한 제공

→ 별도의 검색엔진을 사용하는 것이 아니라, DBMS의 조건절 활용

```
from rest_framework.filters import SearchFilter, OrderingFilter
```

```
class PostModelViewSet(viewsets.ModelViewSet):
```

```
    queryset = Post.objects.all()
```

```
    serializer_class = PostSerializer
```

```
    filter_backends = [SearchFilter, OrderingFilter]
```

```
    search_fields = ['message']    # ?search=      → QuerySet 조건 절에 추가할 필드 지정. 모델 필드 중에 문자열 필드만을 지정.
```

```
    ordering_fields = ['id']       # ?ordering=   → 정렬을 허용할 필드의 화이트 리스트. 미지정 시에 serializer_class에 지정된 필드들.
```

```
    ordering = ['id']             # 디폴트 정렬을 지정
```

참고: django-filter 라이브러리에서는 DRF와의 통합을 지원  
→ `django_filters.rest_framework.DjangoFilterBackend`

`search_fields`    `search_fields = ['=username', '=email']`

## 문자열 패턴 지정

"^" : Starts-with search

"=" : Exact matches

"@" : Full-text search. (단어/구문에 대한 검색, 2020년 2월 현재 장고의 MySQL 백엔드에서만 지원)

"\$" : Regex search

`get_search_fields` 함수로도 구현 가능

인생은 짧습니다.  
파이썬/장고를 쓰세요.

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

Ask Company