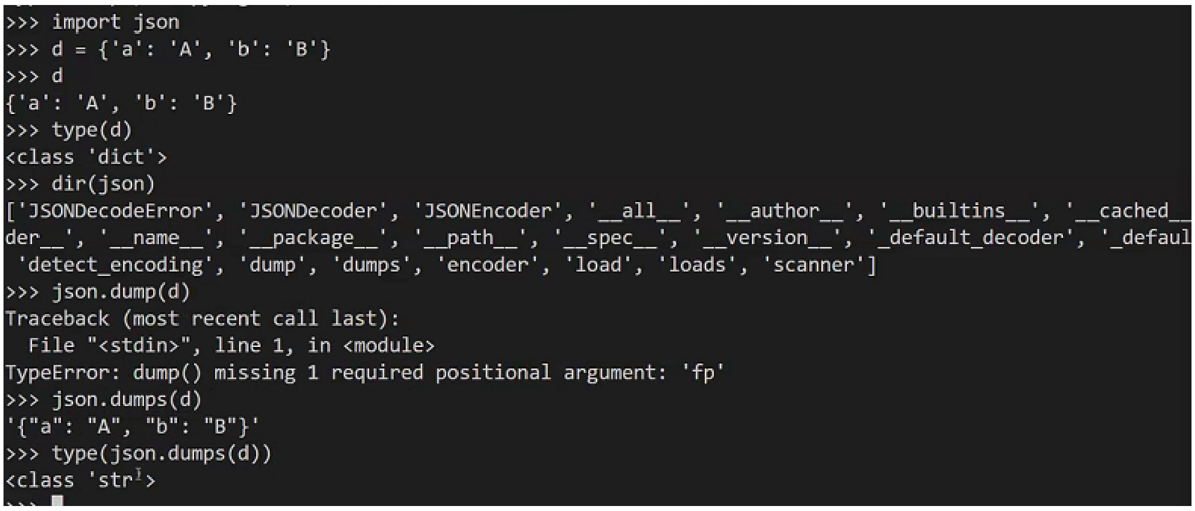
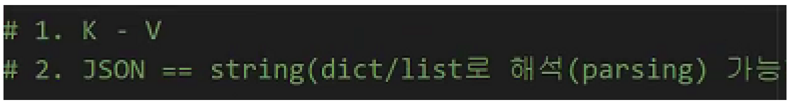


```
>>> import json
>>> d = {'a': 'A', 'b': 'B'}
>>> d
{'a': 'A', 'b': 'B'}
>>> type(d)
<class 'dict'>
>>> dir(json)
['JSONDecodeError', 'JSONDecoder', 'JSONEncoder', '__all__', '__author__', '__builtins__', '__cached__
der__', '__name__', '__package__', '__path__', '__spec__', '__version__', '_default_decoder', '_defaul
 'detect_encoding', 'dump', 'dumps', 'encoder', 'load', 'loads', 'scanner']
>>> json.dump(d)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: dump() missing 1 required positional argument: 'fp'
>>> json.dumps(d)
'{"a": "A", "b": "B"}'
>>> type(json.dumps(d))
<class 'str'>
>>>
```

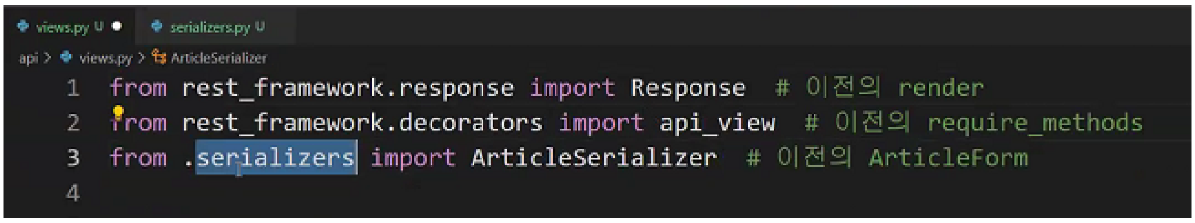
json은 dict이 아니고 str이다

양식이 있는 문서

큰따옴표로 이루어져있는게 json의 정석(작은 따옴표 안됨)

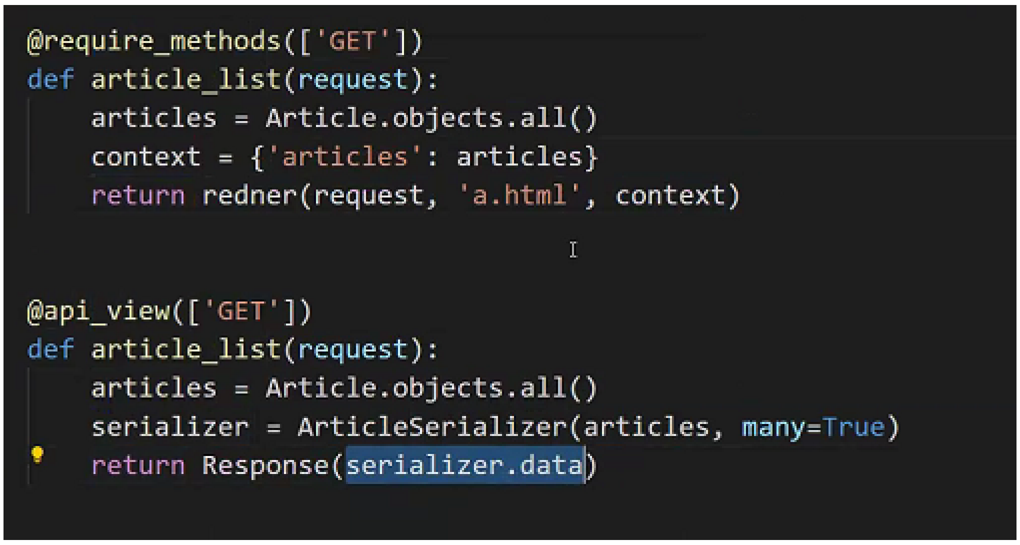


```
# 1. K - V
# 2. JSON == string(dict/list로 해석(parsing) 가능
```



```
from rest_framework.response import Response  # 이전의 render
from rest_framework.decorators import api_view  # 이전의 require_methods
from .serializers import ArticleSerializer  # 이전의 ArticleForm

```

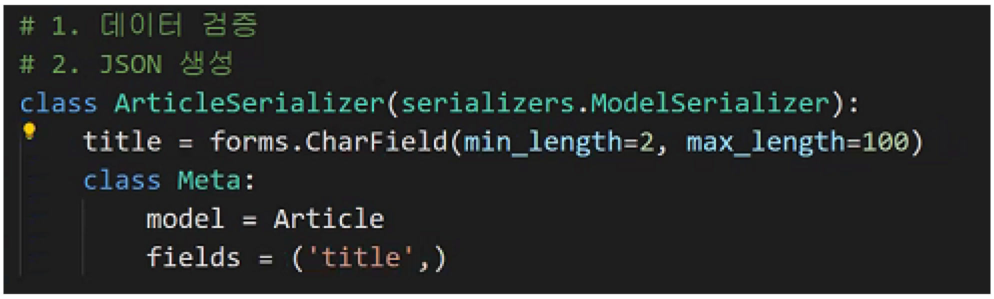
두개 유사



```
@require_methods(['GET'])
def article_list(request):
    articles = Article.objects.all()
    context = {'articles': articles}
    return redner(request, 'a.html', context)


@api_view(['GET'])
def article_list(request):
    articles = Article.objects.all()
    serializer = ArticleSerializer(articles, many=True)
    return Response(serializer.data)
```

serializer로 어떤 것을 보고 내보낼지 결정

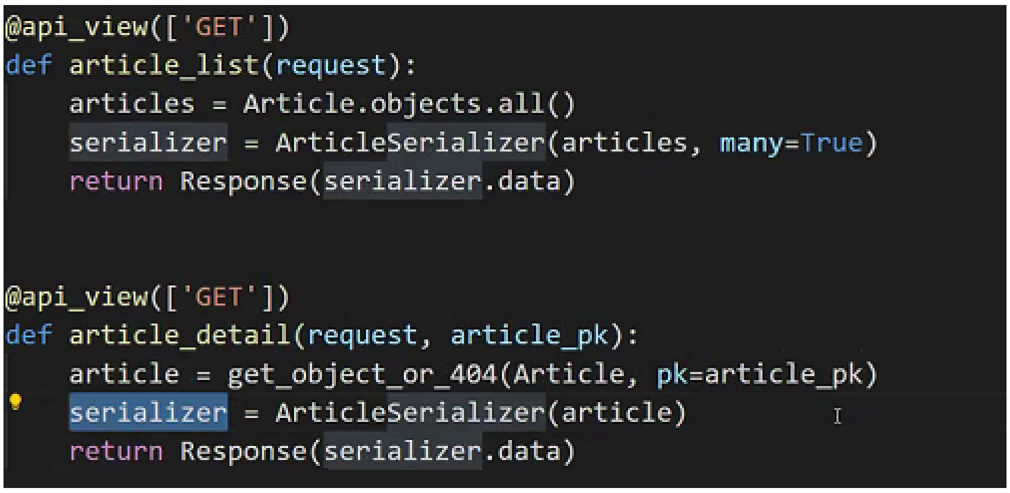
fields 이렇게 하면 제목만



```python
# 1. 데이터 검증
# 2. JSON 생성
class ArticleSerializer(serializers.ModelSerializer):
    title = forms.CharField(min_length=2, max_length=100)
    class Meta:
        model = Article
        fields = ('title',)
```

many=true 는 여러개

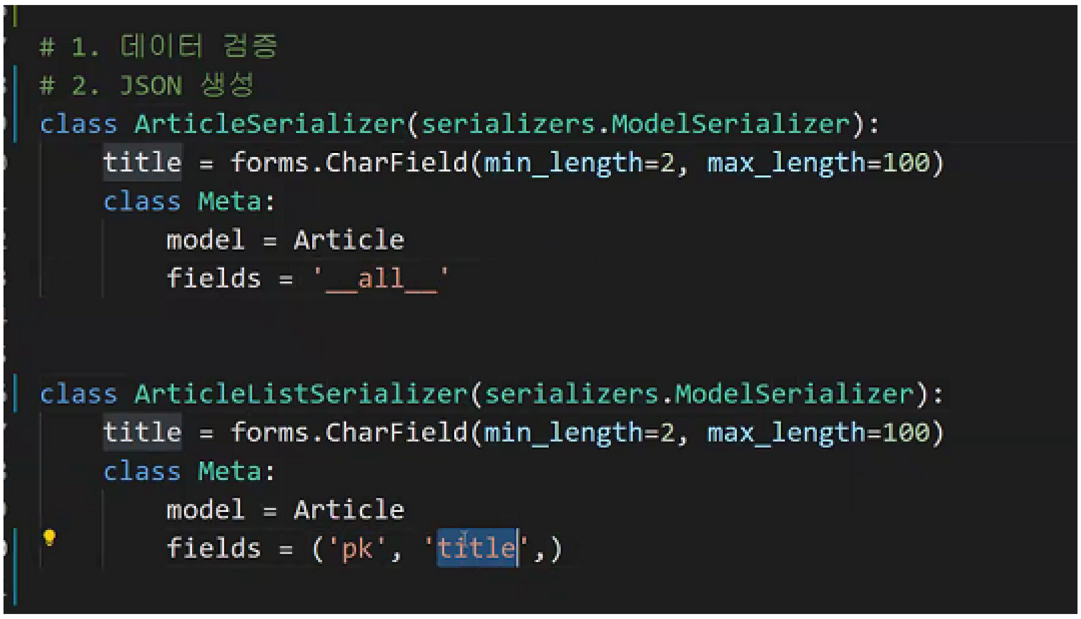
article만 쓰면 한개



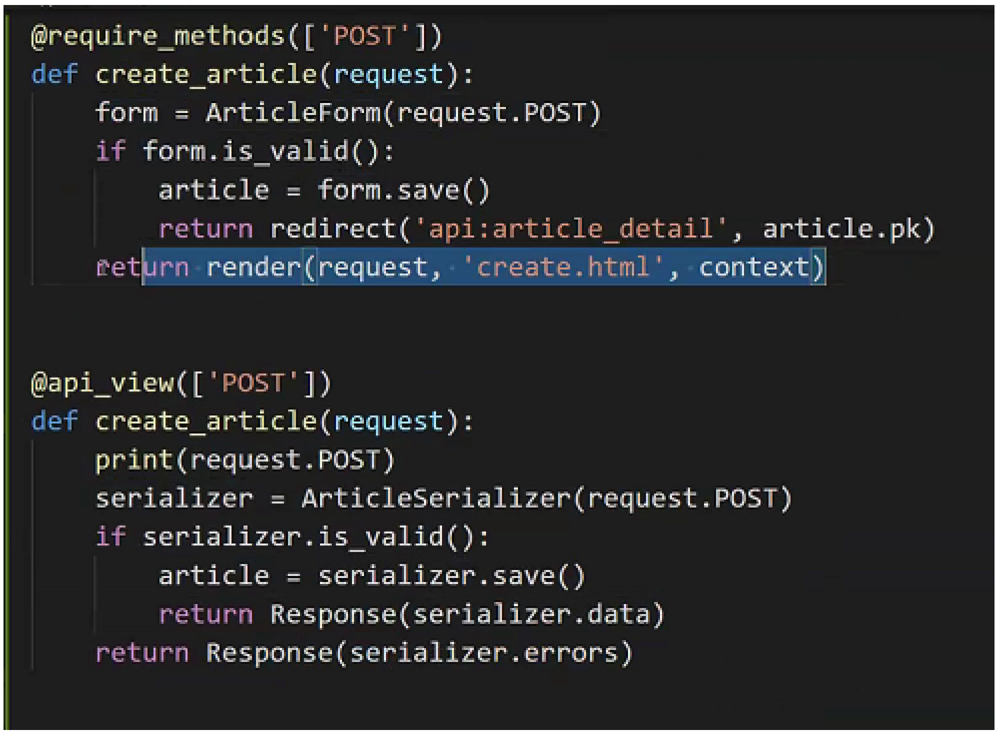
```python
@api_view(['GET'])
def article_list(request):
    articles = Article.objects.all()
    serializer = ArticleSerializer(articles, many=True)
    return Response(serializer.data)


@api_view(['GET'])
def article_detail(request, article_pk):
    article = get_object_or_404(Article, pk=article_pk)
    serializer = ArticleSerializer(article)
    return Response(serializer.data)
```

위는 단일조회용

밑은 목록용(pk와 타이틀만 보기)

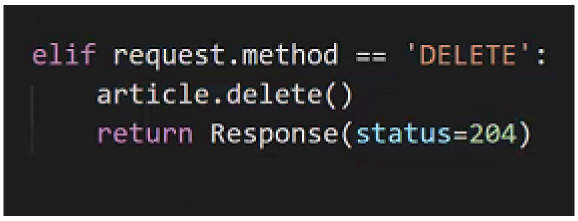


```python
# 1. 데이터 검증
# 2. JSON 생성
class ArticleSerializer(serializers.ModelSerializer):
    title = forms.CharField(min_length=2, max_length=100)
    class Meta:
        model = Article
        fields = '__all__'


class ArticleListSerializer(serializers.ModelSerializer):
    title = forms.CharField(min_length=2, max_length=100)
    class Meta:
        model = Article
        fields = ('pk', 'title',)
```



```python
@require_methods(['POST'])
def create_article(request):
    form = ArticleForm(request.POST)
    if form.is_valid():
        article = form.save()
        return redirect('api:article_detail', article.pk)
    return render(request, 'create.html', context)


@api_view(['POST'])
def create_article(request):
    print(request.POST)
    serializer = ArticleSerializer(request.POST)
    if serializer.is_valid():
        article = serializer.save()
        return Response(serializer.data)
    return Response(serializer.errors)
```

데코레이터 덕분에 들어오는 데이터 json아니면 오류 발생



```python
elif request.method == 'DELETE':
    article.delete()
    return Response(status=204)
```