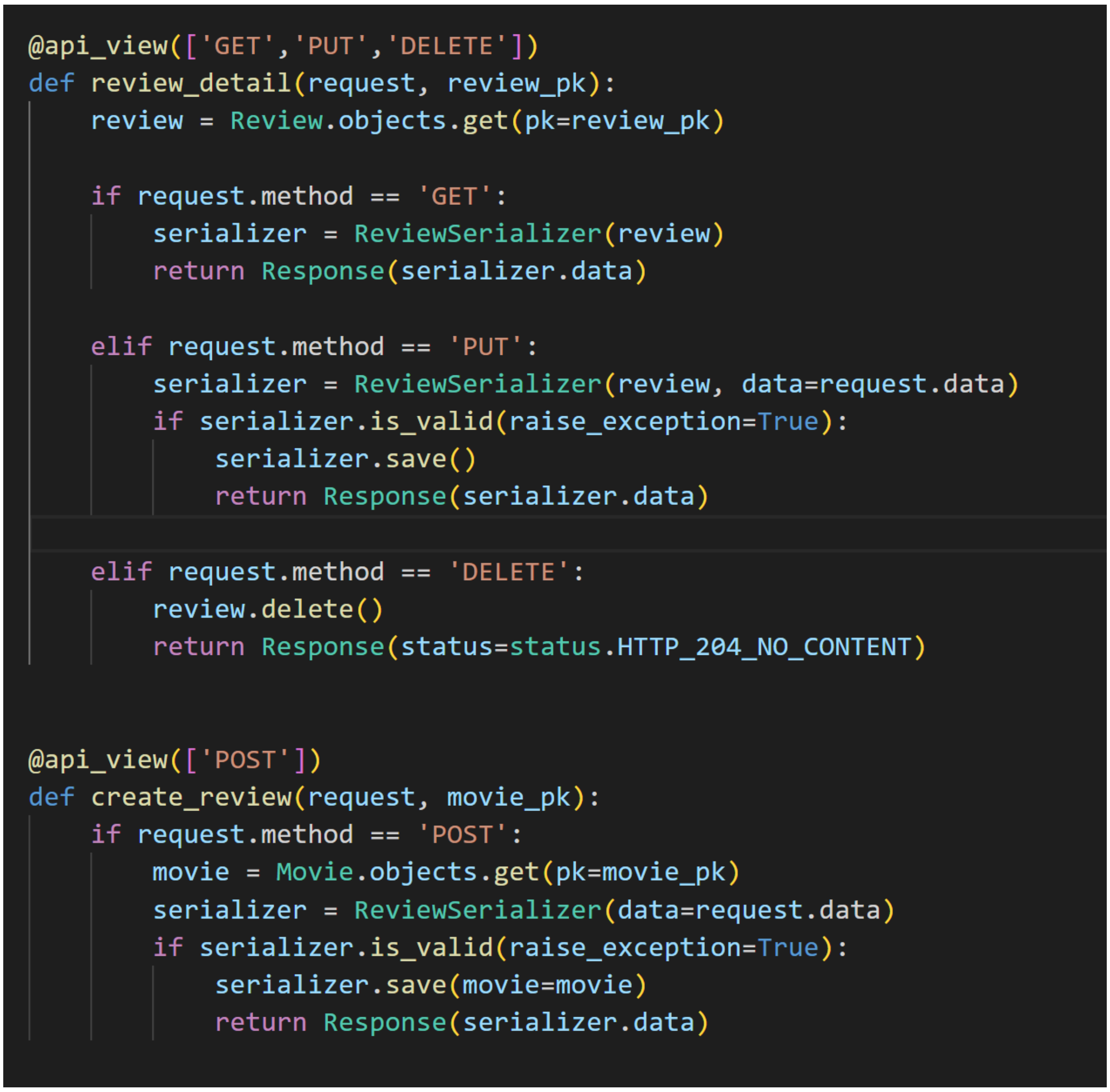
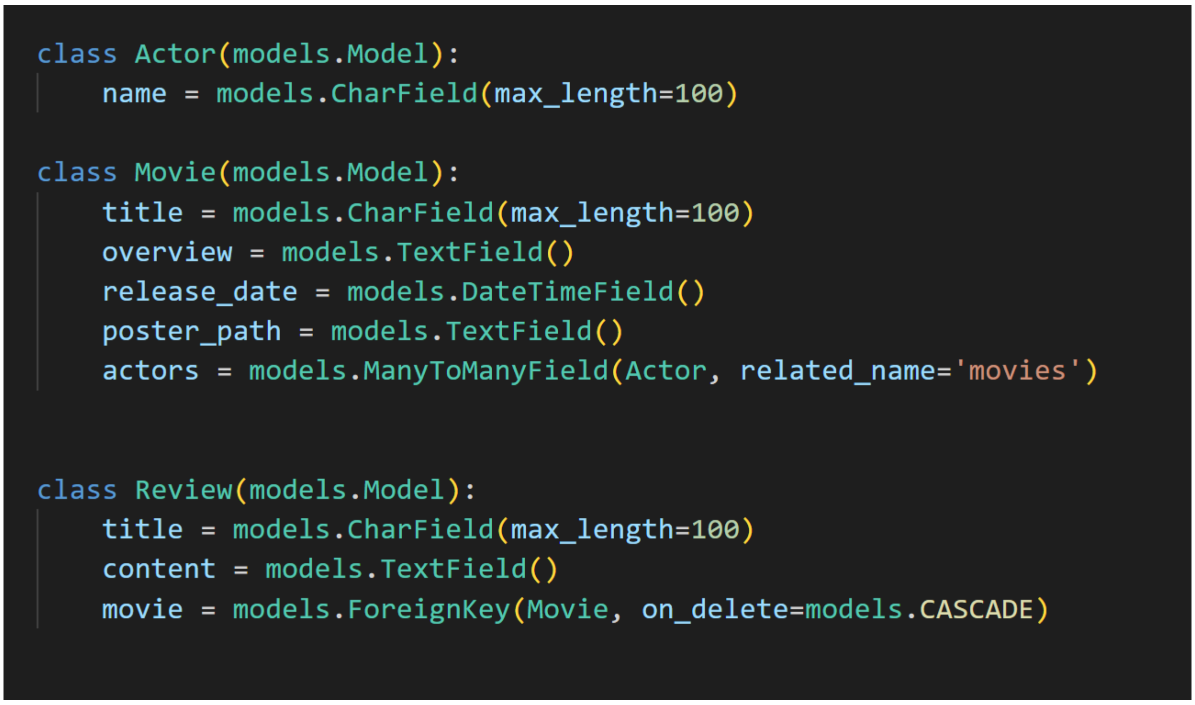
VIEW.py



```python
@api_view(['GET','PUT','DELETE'])
def review_detail(request, review_pk):
    review = Review.objects.get(pk=review_pk)

    if request.method == 'GET':
        serializer = ReviewSerializer(review)
        return Response(serializer.data)

    elif request.method == 'PUT':
        serializer = ReviewSerializer(review, data=request.data)
        if serializer.is_valid(raise_exception=True):
            serializer.save()
            return Response(serializer.data)

    elif request.method == 'DELETE':
        review.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)


@api_view(['POST'])
def create_review(request, movie_pk):
    if request.method == 'POST':
        movie = Movie.objects.get(pk=movie_pk)
        serializer = ReviewSerializer(data=request.data)
        if serializer.is_valid(raise_exception=True):
            serializer.save(movie=movie)
            return Response(serializer.data)
```

Models.py



```python
class Actor(models.Model):
    name = models.CharField(max_length=100)

class Movie(models.Model):
    title = models.CharField(max_length=100)
    overview = models.TextField()
    release_date = models.DateTimeField()
    poster_path = models.TextField()
    actors = models.ManyToManyField(Actor, related_name='movies')


class Review(models.Model):
    title = models.CharField(max_length=100)
    content = models.TextField()
    movie = models.ForeignKey(Movie, on_delete=models.CASCADE)
```

Serializers.py

```python
# Actor of Detail 을 위한 serializer
class ActorDetailofMovielist(serializers.ModelSerializer):

    class Meta:
        model = Movie
        fields = ('title',)


# Movie of Detail 을 위한 serializer - actor
class MovieDetailofActorlist(serializers.ModelSerializer):

    class Meta:
        model = Actor
        fields = ('name',)


# Actor Detail
class ActorDetailSerializer(serializers.ModelSerializer):
    movies = ActorDetailofMovielist(many=True, read_only=True)

    class Meta:
        model = Actor
        fields = ('id', 'movies', 'name',)
        # read_only_fields = ('movies')


# Movie Detail
class MovieDetailSerializer(serializers.ModelSerializer):
    actors = MovieDetailofActorlist(many=True, read_only=True)
    review = ReviewSerializer(many=True, read_only=True ,source='review_set')

    class Meta:
        model = Movie
        # fields = ('id', 'actors', 'review_set','title', 'overview','release_date', 'poster_path')
        fields = '__all__'
```