Models.py

# myapp/models.py

from django.db import models

from django.conf import settings

User = settings.AUTH\_USER\_MODEL

class produitBD(models.Model):

# user = models.ForeignKey(User, null=True, on\_delete=models.SET\_NULL)

nom = models.CharField(max\_length=50, null=True)

content = models.TextField()

prix = models.DecimalField(max\_digits=15, decimal\_places=2)

def \_\_str\_\_(self):

return self.nom

class Auteur(models.Model):

nom = models.CharField(max\_length=100)

nationalite = models.CharField(max\_length=100)

def \_\_str\_\_(self):

return self.nom

class Livre(models.Model):

titre = models.CharField(max\_length=200)

auteur = models.ForeignKey(Auteur, on\_delete=models.CASCADE)

date\_publication = models.DateField()

created\_by = models.ForeignKey(User, on\_delete=models.SET\_NULL, null=True, blank=True)

def \_\_str\_\_(self):

return self.titre

Serializer.py

from rest\_framework import serializers

from .models import produitBD, Livre

from .validetors import validate\_name

class produitSerializer(serializers.ModelSerializer):

# user\_name = serializers.CharField(source="user.username", read\_only=True)

nom = serializers.CharField(validators = [validate\_name])

class Meta:

model = produitBD

fields = ['pk', 'nom', 'content', 'prix'] # Retirez 'pk' de cette liste

# ici on va afficher le nom de l'utilisateyr qui a cree le vivre

class LivreSerializer(serializers.ModelSerializer):

# auteur = AuteurSerializer()

created\_by = serializers.CharField()

class Meta:

model = Livre

fields = ('id', 'titre', 'auteur', 'date\_publication', 'created\_by')

Views.py

from rest\_framework import viewsets, authentication, mixins

from .permissions import prouitPermissions

from .models import produitBD, Livre

from .serializer import produitSerializer, LivreSerializer

class produitViewsets(viewsets.ModelViewSet):

queryset = produitBD.objects.all()

serializer\_class = produitSerializer

authentication\_classes = [authentication.SessionAuthentication]

permission\_classes = [prouitPermissions]

class produitViewsetMixins(viewsets.GenericViewSet, mixins.RetrieveModelMixin, mixins.ListModelMixin,):

queryset = produitBD.objects.all()

serializer\_class = produitSerializer

authentication\_classes = [authentication.SessionAuthentication]

permission\_classes = [prouitPermissions]

class LivreViewSet(viewsets.ModelViewSet):

queryset = Livre.objects.all()

serializer\_class = LivreSerializer

Admin.py

from django.contrib import admin

from .models import produitBD, Livre,Auteur

# Register your models here.

admin.site.register(produitBD)

admin.site.register(Livre)

admin.site.register(Auteur)

Permission.py

from django.contrib import admin

from .models import produitBD, Livre,Auteur

# Register your models here.

admin.site.register(produitBD)

admin.site.register(Livre)

admin.site.register(Auteur)

Validators.py

from rest\_framework import serializers

from .models import produitBD

def validate\_name(value):

queryset = produitBD.objects.filter(nom\_\_iexact=value)

if queryset.exists():

raise serializers.ValidationError(f"Le produit {value} existe déjà")

return value

Routers.py

from rest\_framework.routers import DefaultRouter

from produit.views import produitViewsets, produitViewsetMixins, LivreViewSet

router = DefaultRouter()

router.register('produit-1', produitViewsets, basename='produit-2')

router.register('produitMixins-1', produitViewsetMixins, basename='produitMixins-2')

router.register('livre-1', LivreViewSet, basename='livre-2')

urlpatterns = router.urls