Models

from django.db import models

class Breed(models.Model):

    name = models.TextField("Название породы")

    class Meta:

        verbose\_name = "Порода"

        verbose\_name\_plural = "Породы"

    def \_\_str\_\_(self) -> str:

        return self.name

# Create your models here.

class Dog(models.Model):

    name = models.TextField("Название")

    breed = models.ForeignKey("Breed", on\_delete=models.CASCADE, null=True)

    class Meta:

        verbose\_name = "Собака"

        verbose\_name\_plural = "Собаки"

    def \_\_str\_\_(self) -> str:

        return self.name

class Owner(models.Model):

    first\_name = models.TextField("Имя")

    last\_name = models.TextField("Фамилия")

    phone\_number = models.TextField("Номер телефона")

    dogs = models.ForeignKey("Dog", on\_delete=models.CASCADE, null=True)

    class Meta:

        verbose\_name = "Владелец"

        verbose\_name\_plural = "Владельцы"

    def \_\_str\_\_(self) -> str:

        return f"{self.first\_name} {self.last\_name}"

class WeightEntry(models.Model):

    date = models.DateField("Дата")

    weight = models.TextField("Вес (кг)")

    dogs = models.ForeignKey("Dog", on\_delete=models.CASCADE, null=True)

    class Meta:

        verbose\_name = "Запись веса"

        verbose\_name\_plural = "Записи веса"

    def \_\_str\_\_(self):

        return f"{self.weight} кг"

class Vaccination(models.Model):

    name = models.TextField("Название прививки")

    date = models.DateField("Дата прививки")

    dogs = models.ForeignKey("Dog", on\_delete=models.CASCADE, null=True)

    class Meta:

        verbose\_name = "Прививка"

        verbose\_name\_plural = "Прививки"

    def \_\_str\_\_(self):

        return f"{self.name}"

Serializers

from rest\_framework import serializers

from dogs.models import Breed, Dog, Owner, WeightEntry, Vaccination

class BreedSerializer(serializers.ModelSerializer):

    class Meta:

        model = Breed

        fields = "\_\_all\_\_"

class DogSerializer(serializers.ModelSerializer):

    breed = BreedSerializer()

    class Meta:

        model = Dog

        fields = ['id', 'name', 'breed']

class OwnerSerializer(serializers.ModelSerializer):

    owner = DogSerializer()

    class Meta:

        model = Owner

        fields = ['id', 'first\_name', 'last\_name', 'phone\_number']

class WeightEntrySerializer(serializers.ModelSerializer):

    weight = DogSerializer()

    class Meta:

        model = WeightEntry

        fields = ['id', 'date', 'weight']

class VaccinationSerializer(serializers.ModelSerializer):

    vac = DogSerializer()

    class Meta:

        model = Vaccination

        fields = ['id', 'name', 'date']

Admin

from django.contrib import admin

from dogs.models import Vaccination, Breed, Dog, Owner, WeightEntry

# Register your models here.

@admin.register(Dog)

class DogAdmin(admin.ModelAdmin):

    list\_display = ['id', 'name', 'breed\_\_id']

@admin.register(Breed)

class BreedAdmin(admin.ModelAdmin):

    list\_display = ['id', 'name']

@admin.register(Owner)

class OwnerAdmin(admin.ModelAdmin):

    list\_display = ['id', 'first\_name', 'last\_name', 'phone\_number']

@admin.register(WeightEntry)

class WeightEntryAdmin(admin.ModelAdmin):

    list\_display = ['id', 'date', 'weight']

@admin.register(Vaccination)

class VaccinationAdmin(admin.ModelAdmin):

    list\_display = ['id', 'name', 'date']

'rest\_framework',

    'dogs',

from django.contrib import admin

from django.urls import include, path

from dogs import views

from rest\_framework.routers import DefaultRouter

from dogs.api import DogsViewset

router = DefaultRouter()

router.register("dogs", DogsViewset, basename="dogs")

urlpatterns = [

    path('', views.ShowDogsView.as\_view()),

    path('admin/', admin.site.urls),

    path('api/', include(router.urls)),

]

django==5.1

djangorestframework==3.15.2

api

from rest\_framework.viewsets import GenericViewSet

from rest\_framework import mixins, viewsets

from dogs.models import Dog

from dogs.serializers import DogSerializer

class DogsViewset(mixins.ListModelMixin, GenericViewSet):

    queryset = Dog.objects.all()

    serializer\_class = DogSerializer

views

from typing import Any, Dict

from django.http import HttpResponse

from django.shortcuts import render

from django.views.generic import TemplateView

from dogs.models import Dog

class ShowDogsView(TemplateView):

    template\_name = "dogs/show\_students.html"

    def get\_context\_data(self, \*\*kwargs: Any) -> Dict[str, Any]:

        context = super().get\_context\_data(\*\*kwargs)

        context['dogs'] = Dog.objects.all()

        return context

# class ShowDogsView(View):

#     def get(request, \*args \*\*kwargs):

#         dogs = Dog.objects.all()

#         result = ""

#         for s in dogs:

#             result += s.name + "<br>"

#         return HttpResponse(result)

# # Create your views here.

# def show\_dogs(request):

#     return HttpResponse("Привет")