Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»
Отчет по РК №2

Выполнил: Студент группы ИУ5-53Б
Борисов Андрей Михайлович

Москва - 2021

Условия рубежного контроля №2 по курсу РИП

Рубежный контроль представляет собой разработку веб-приложения с использованием фреймворка Django. Веб-приложение должно выполнять следующие функции:

1. Создайте проект Python Django с использованием стандартных средств Django.
2. Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

Текст программы

Models.py

```python
from django.db import models


class Streets(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=20)

    class Meta:
        db_table = 'streets'


class Houses(models.Model):
    id = models.IntegerField(primary_key=True)
    number = models.IntegerField()
    floor = models.IntegerField()
    street_id = models.ForeignKey(Streets, on_delete=models.PROTECT)

    class Meta:
        db_table = 'houses'
```

Serializers.py

```python
from rest_framework import serializers

from simple_app.models import Houses, Streets


class HouseSerializer(serializers.ModelSerializer):
    class Meta:

        model = Houses
        fields = "__all__"


class StreetSerializer(serializers.ModelSerializer):
    class Meta:

        model = Streets
        fields = "__all__"
```

Views.py

```python
from django.shortcuts import render
from rest_framework import viewsets

from simple_app.models import Streets, Houses
from simple_app.serializers import StreetSerializer, HouseSerializer


class StreetViewSet(viewsets.ModelViewSet):
    queryset = Streets.objects.all()
    serializer_class = StreetSerializer


class HouseViewSet(viewsets.ModelViewSet):
    queryset = Houses.objects.all()
    serializer_class = HouseSerializer


def report(request):
    return render(request, 'report.html', {'data': {
        'houses': Houses.objects.select_related('street_id')
    }})
```

Urls.py

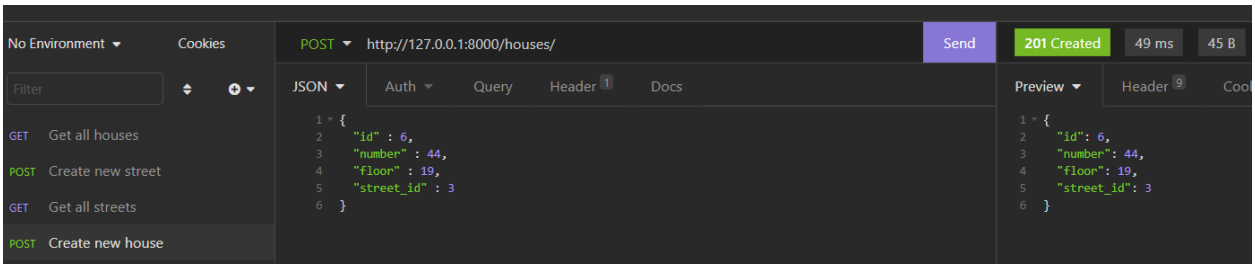
```python
from rest_framework import routers
from django.urls import path, include
from django.contrib import admin
from django.urls import path
from simple_app import views as views

router = routers.DefaultRouter()
router.register('streets', views.StreetViewSet)
router.register('houses', views.HouseViewSet)

urlpatterns = [

    path('', include(router.urls)),
    path('report/', views.report),
    path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),
    path('admin/', admin.site.urls),
]
```

Settings.py

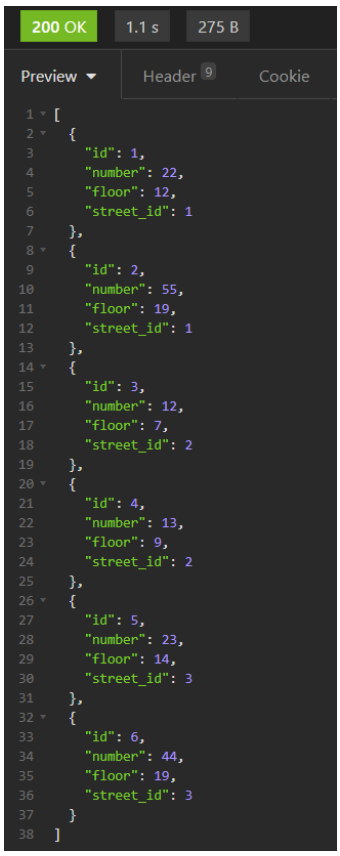
```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'rk2',
        'USER': 'postgres',
        'PASSWORD': '6146534',
        'HOST': 'localhost',
        'PORT': 5432,
    }
}
```

Работа приложения

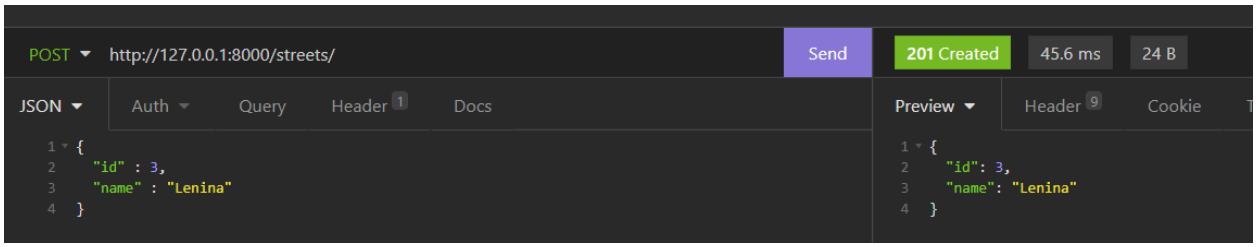
Создание нового дома POST метод:



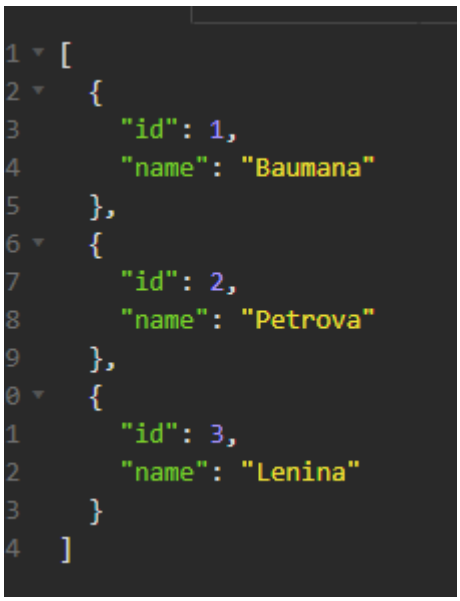
Запрос списка всех домов GET метод:



Создание новой улицы POST метод:



Запрос списка всех улиц GET метод:



Соединение данных из двух таблиц:

Список Домов:

------- **Дом 22** -------

- Этажность: *12*
- Улица: *Baumana*

------- **Дом 55** -------

- Этажность: *19*
- Улица: *Baumana*

------- **Дом 12** -------

- Этажность: *7*
- Улица: *Petrova*

------- **Дом 13** -------

- Этажность: *9*
- Улица: *Petrova*

------- **Дом 23** -------

- Этажность: *14*
- Улица: *Lenina*

------- **Дом 44** -------

- Этажность: *19*
- Улица: *Lenina*