

**Московский государственный технический
университет им. Н. Э. Баумана**

Отчёт по лабораторной работе №6 по курсу «Разработка интернет
приложений».

«Разработка REST API с использованием Django REST Framework».

Выполнил:
Анцифров Н. С.
студент группы ИУ5-51Б

Проверил:
Гапанюк Ю. Е.

Подпись и дата:

Подпись и дата:

Москва, 2021 г.

Содержание

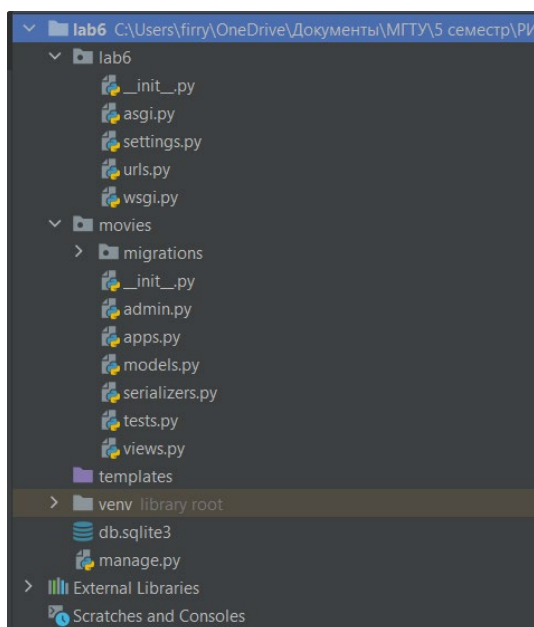
1. Задание лабораторной работы.	2
2. Текст программы.....	2
3. Выполнение программы.....	6

1. Задание лабораторной работы.

С использованием Django REST Framework разработайте REST API для одной модели (одной таблицы базы данных).

2. Текст программы.

Структура проекта:



settings.py

```
"""
Django settings for lab6 project.

Generated by 'django-admin startproject' using Django 4.0.

For more information on this file, see
https://docs.djangoproject.com/en/4.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.0/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent
```

```

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-
h%sf(n#xgqf=@ail9lg6r)05^@dls)eyuhghv00qp!!283(d84'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # DRF
    'rest_framework',

    # Наше приложение
    'movies',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'lab6.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates']
        ,
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'lab6.wsgi.application'

# Database
# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

```

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/4.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

urls.py

```

from django.contrib import admin
from movies import views as movies_views
from django.urls import include, path
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'movies', movies_views.MovieViewSet)

```

```
# Wire up our API using automatic URL routing.
# Additionally, we include login URLs for the browsable API.
urlpatterns = [
    path('', include(router.urls)),
    path('api-auth/', include('rest_framework.urls',
namespace='rest_framework')),
    path('admin/', admin.site.urls),
]
```

models.py

```
from django.db import models

class Movie(models.Model):
    name = models.CharField(max_length=50, verbose_name="Название фильма")
    description = models.CharField(max_length=255, verbose_name="Описание
фильма")
    rating = models.DecimalField(max_digits=3, decimal_places=1,
verbose_name="Рейтинг фильма")
    is_growing = models.BooleanField(verbose_name="Увеличивается ли рейтинг
фильма?")
    date_modified = models.DateTimeField(auto_now=True,
verbose_name="Когда последний раз
обновлялась информация о фильме?")
```

serializers.py

```
from movies.models import Movie
from rest_framework import serializers

class MovieSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = Movie
        # Поля, которые мы сериализуем
        fields = ["pk", "name", "description", "rating", "is_growing",
"date_modified"]
```

views.py

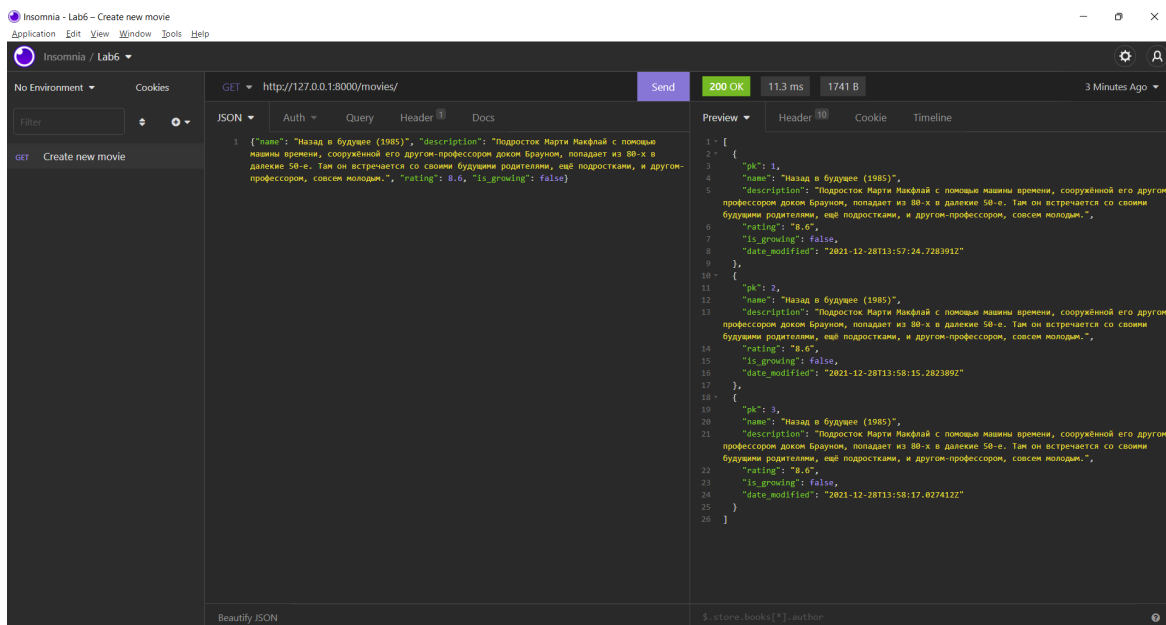
```
from rest_framework import viewsets
from movies.serializers import MovieSerializer
from movies.models import Movie

class MovieViewSet(viewsets.ModelViewSet):
    """
    API endpoint, который позволяет просматривать и редактировать акции
компаний
    """
    # queryset всех пользователей для фильтрации по дате последнего изменения
    queryset = Movie.objects.all().order_by('date_modified')
    serializer_class = MovieSerializer # Сериализатор для модели
```

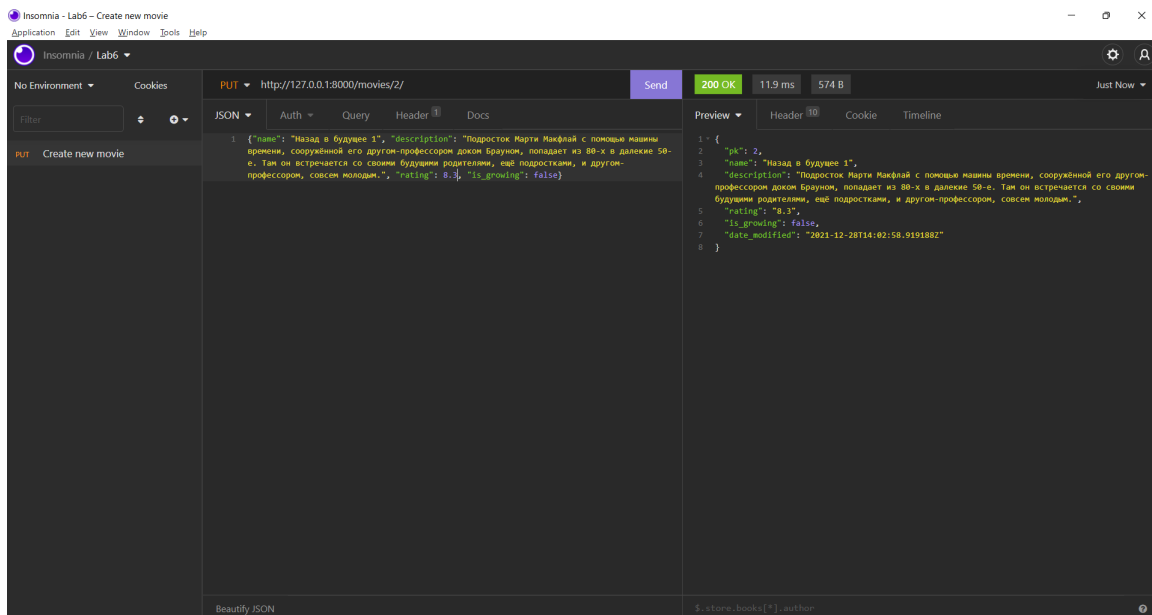
3. Выполнение программы.

Запустим сервер и проверим правильность работы API с помощью Insomnia:

Создадим запрос «Create new movie», добавим несколько записей через метод POST, проверим работу с помощью метода GET:



Изменим данные у объекта с pk=2 с помощью метода PUT:



Выведем все объекты с помощью GET:

```

1 - [
2   {
3     "pk": 1,
4     "name": "Назад в будущее (1985)",
5     "description": "Подросток Марти Макфлай с помощью машины времени, сооружённой его другом-
6     профессором доком Брауном, попадает из 80-х в далекие 50-е. Там он встречается со своими
7     будущими родителями, ещё подростками, и другом-профессором, совсем молодым.",
8     "rating": "8.6",
9     "is_growing": false,
10    "date_modified": "2021-12-28T13:57:24.728391Z"
11  },
12  {
13    "pk": 3,
14    "name": "Назад в будущее (1985)",
15    "description": "Подросток Марти Макфлай с помощью машины времени, сооружённой его другом-
16    профессором доком Брауном, попадает из 80-х в далекие 50-е. Там он встречается со своими
17    будущими родителями, ещё подростками, и другом-профессором, совсем молодым.",
18    "rating": "8.6",
19    "is_growing": false,
20    "date_modified": "2021-12-28T13:58:17.027412Z"
21  },
22  {
23    "pk": 2,
24    "name": "Назад в будущее 1",
25    "description": "Подросток Марти Макфлай с помощью машины времени, сооружённой его другом-
26    профессором доком Брауном, попадает из 80-х в далекие 50-е. Там он встречается со своими
27    будущими родителями, ещё подростками, и другом-профессором, совсем молодым.",
28    "rating": "8.3",
29    "is_growing": false,
30    "date_modified": "2021-12-28T14:02:58.919188Z"
31  }
32 ]

```

Попробуем удалить объекты с названием «Назад в будущее (1985)» с помощью метода DELETE:

DELETE ▼ http://127.0.0.1:8000/movies/1/
Send

DELETE ▼ http://127.0.0.1:8000/movies/3/
Send

Выведем снова все объекты с помощью GET и убедимся в выполнении предыдущих действий по методу DELETE:

GET ▼ http://127.0.0.1:8000/movies/
Send
200 OK
4.29 ms
576 B
Just Now ▼

JSON ▼
Auth ▼
Query
Header 11
Docs

Preview ▼
Header 10
Cookie
Timeline

1 { "name": "Назад в будущее 1", "description": "Подросток Марти Макфлай с помощью машины
2 времени, сооружённой его другом-профессором доком Брауном, попадает из 80-х в далекие 50-
3 е. Там он встречается со своими будущими родителями, ещё подростками, и другом-
4 профессором, совсем молодым.", "rating": 8.3, "is_growing": false }
5
6
7
8
9
10]