Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»



**Отчет**

**Лабораторная работа № 6**

**По курсу «Разработка интернет приложений»**

#### Тема работы: " Разработка REST API с использованием Django REST Framework "

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5Ц-74Б

Семенова А.А.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2021 г.

Москва 2021

**СОДЕРЖАНИЕ**

[1. Описание задания 3](#_Toc88941587)

[2. Текст программы 4](#_Toc88941588)

[3. Результаты выполнения программы 9](#_Toc88941589)

# Описание задания

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

# Текст программы

**animals/admin.py**

from django.contrib import admin  
from .models import Animal  
  
*# Register your models here.*admin.site.register(Animal)

**animals/apps.py**

from django.apps import AppConfig  
  
  
class AnimalsConfig(AppConfig):  
 default\_auto\_field = 'django.db.models.BigAutoField'  
 name = 'animals'

**animals/models.py**

from django.db import models  
  
  
*# Create your models here.*class Animal(models.Model):  
 animal\_type = models.CharField('Вид животного', max\_length=50)  
 animal\_name = models.CharField('Кличка животного', max\_length=50)  
 animal\_photo = models.URLField('Фото жифотного', max\_length=200)  
  
 def \_\_str\_\_(self):  
 return f'Тип: {self.animal\_type}; Имя: {self.animal\_name}; URL фото: {self.animal\_photo}'  
  
 class Meta:  
 verbose\_name = 'Животное'  
 verbose\_name\_plural = 'Животные'

**animals/tests.py**

from django.test import TestCase  
  
*# Create your tests here.*

**animals/views.py**

from django.shortcuts import render  
from rest\_framework import viewsets  
from .serializers import AnimalSerializer  
from .models import Animal  
  
  
*# Create your views here.*class AnimalViewSet(viewsets.ModelViewSet):  
 *"""  
 API endpoint, который позволяет просматривать и редактировать акции компаний  
 """  
 # queryset всех пользователей для фильтрации по дате последнего изменения* queryset = Animal.objects.all()  
 serializer\_class = AnimalSerializer *# Сериализатор для модели*

**animals/** **serializers.py**

from rest\_framework import serializers  
from .models import Animal  
  
  
class AnimalSerializer(serializers.ModelSerializer):  
 class Meta:  
 *# Модель, которую мы сериализуем* model = Animal  
 *# Поля, которые мы сериализуем* fields = ["animal\_type", "animal\_name", "animal\_photo"]

**LAB\_6/settings.py**

*"""  
Django settings for LAB\_6 project.  
  
Generated by 'django-admin startproject' using Django 3.2.9.  
  
For more information on this file, see  
https://docs.djangoproject.com/en/3.2/topics/settings/  
  
For the full list of settings and their values, see  
https://docs.djangoproject.com/en/3.2/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/3.2/howto/deployment/checklist/  
  
# SECURITY WARNING: keep the secret key used in production secret!*SECRET\_KEY = 'django-insecure-m#mp=%7jeczg4o83^1(n%qba5i1wk!\_a1yu(^grl(pmba-%t+g'  
  
*# 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',  
 'rest\_framework',  
 'animals',  
]  
  
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 = 'LAB\_6.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 = 'LAB\_6.wsgi.application'  
  
  
*# Database  
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases*DATABASES = {  
 'default': {  
 'ENGINE': 'django.db.backends.sqlite3',  
 'NAME': BASE\_DIR / 'db.sqlite3',  
 }  
}  
  
  
*# Password validation  
# https://docs.djangoproject.com/en/3.2/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/3.2/topics/i18n/*LANGUAGE\_CODE = 'ru-ru'  
  
TIME\_ZONE = 'UTC'  
  
USE\_I18N = True  
  
USE\_L10N = True  
  
USE\_TZ = True  
  
  
*# Static files (CSS, JavaScript, Images)  
# https://docs.djangoproject.com/en/3.2/howto/static-files/*STATIC\_URL = '/static/'  
  
*# Default primary key field type  
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field*DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

**LAB\_6/urls.py**

*"""LAB\_6 URL Configuration  
  
The `urlpatterns` list routes URLs to views. For more information please see:  
 https://docs.djangoproject.com/en/3.2/topics/http/urls/  
Examples:  
Function views  
 1. Add an import: from my\_app import views  
 2. Add a URL to urlpatterns: path('', views.home, name='home')  
Class-based views  
 1. Add an import: from other\_app.views import Home  
 2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')  
Including another URLconf  
 1. Import the include() function: from django.urls import include, path  
 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))  
"""*from django.contrib import admin  
from django.urls import path, include  
from rest\_framework import routers  
from animals import views as animal\_views  
  
router = routers.DefaultRouter()  
router.register(r'animals', animal\_views.AnimalViewSet)  
  
*# 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),  
]

**LAB\_6/wsgi.py**

*"""  
WSGI config for LAB\_6 project.  
  
It exposes the WSGI callable as a module-level variable named ``application``.  
  
For more information on this file, see  
https://docs.djangoproject.com/en/3.2/howto/deployment/wsgi/  
"""*import os  
  
from django.core.wsgi import get\_wsgi\_application  
  
os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'LAB\_6.settings')  
  
application = get\_wsgi\_application()

# Результаты выполнения программы

**Лог**

"C:\Program Files\JetBrains\PyCharm 2021.2\bin\runnerw.exe" "A:\anzh\YandexDisk\МГТУ\7-й семестр\РИП\ЛР\Py\LAB\_6\venv\Scripts\python.exe" "A:/anzh/YandexDisk/МГТУ/7-й семестр/РИП/ЛР/Py/LAB\_6/manage.py" runserver 8000

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

November 27, 2021 - 21:19:59

Django version 3.2.9, using settings 'LAB\_6.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

[27/Nov/2021 21:20:02] "GET / HTTP/1.1" 200 5546

[27/Nov/2021 21:20:02] "GET /static/rest\_framework/js/jquery-3.5.1.min.js HTTP/1.1" 304 0

[27/Nov/2021 21:20:02] "GET /static/rest\_framework/js/ajax-form.js HTTP/1.1" 304 0

[27/Nov/2021 21:20:02] "GET /static/rest\_framework/js/csrf.js HTTP/1.1" 304 0

[27/Nov/2021 21:20:02] "GET /static/rest\_framework/js/bootstrap.min.js HTTP/1.1" 304 0

[27/Nov/2021 21:20:02] "GET /static/rest\_framework/js/prettify-min.js HTTP/1.1" 304 0

[27/Nov/2021 21:20:02] "GET /static/rest\_framework/js/default.js HTTP/1.1" 304 0

[27/Nov/2021 21:20:03] "GET /static/rest\_framework/css/prettify.css HTTP/1.1" 200 817

[27/Nov/2021 21:20:03] "GET /static/rest\_framework/css/bootstrap-tweaks.css HTTP/1.1" 200 3385

[27/Nov/2021 21:20:03] "GET /static/rest\_framework/css/default.css HTTP/1.1" 200 1152

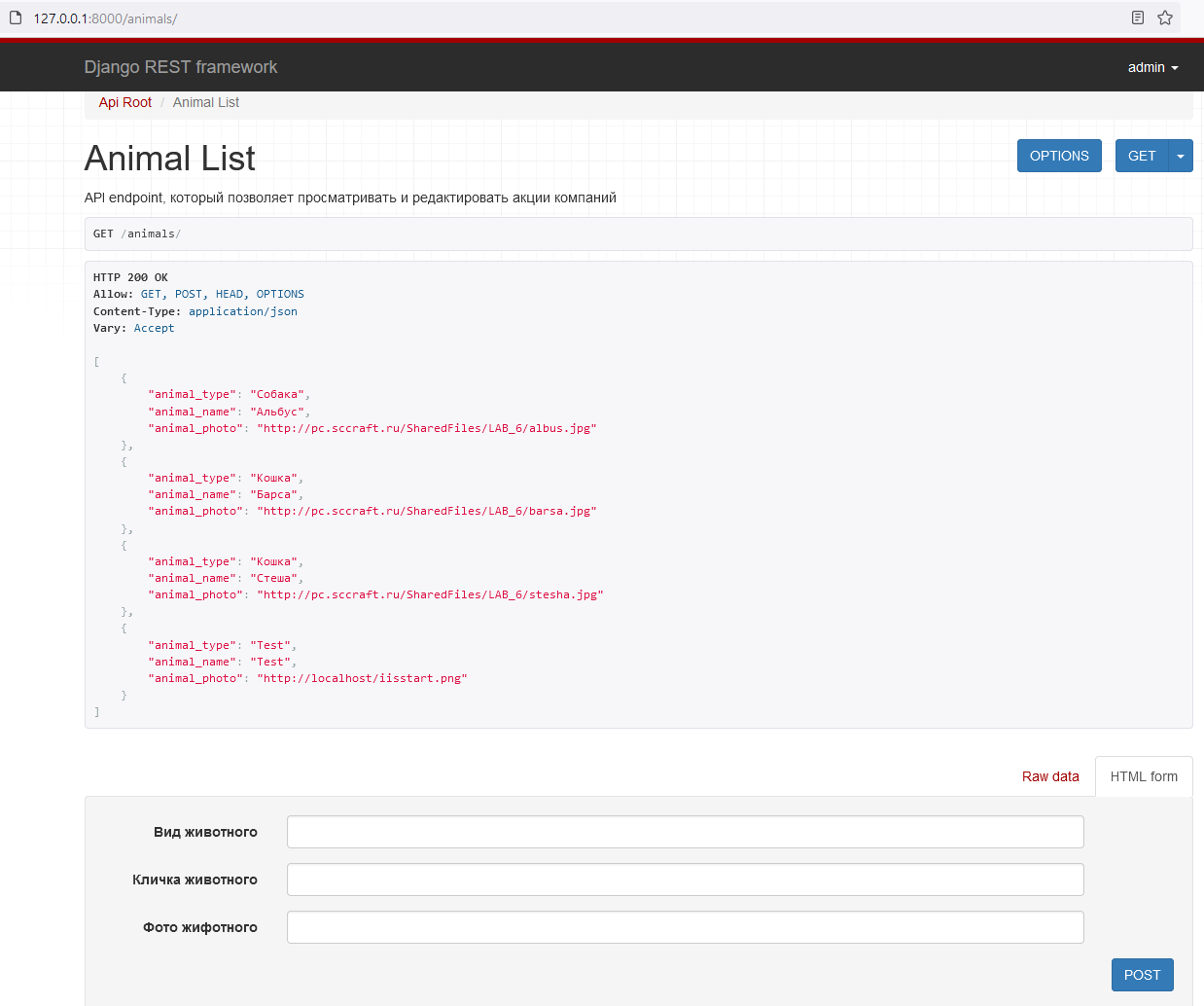
[27/Nov/2021 21:20:03] "GET /static/rest\_framework/css/bootstrap.min.css HTTP/1.1" 200 121457

………………………………………………………………………………………………

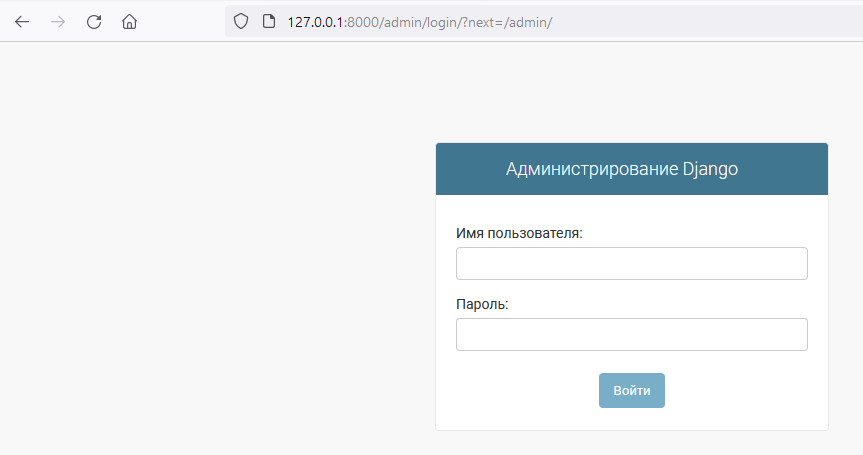
**Главная страница (API root)**

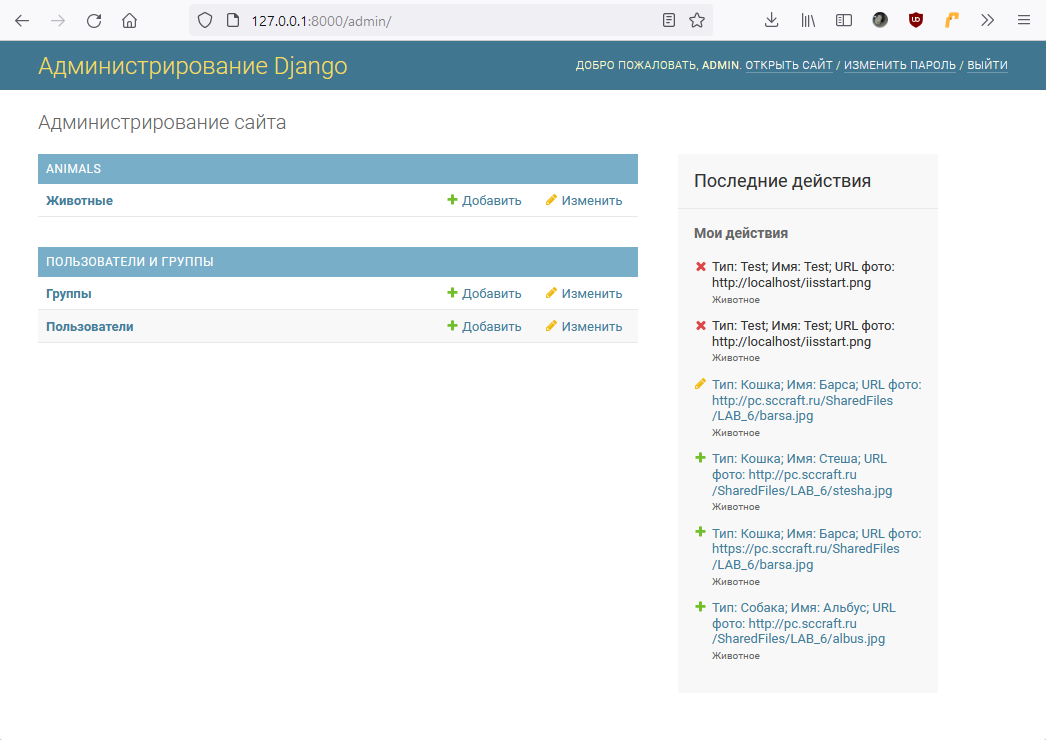


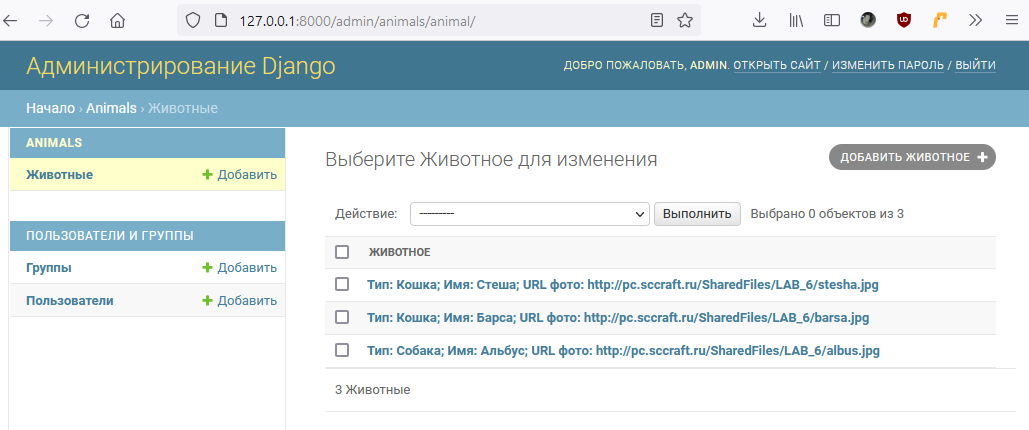
**API животных (/animals)**

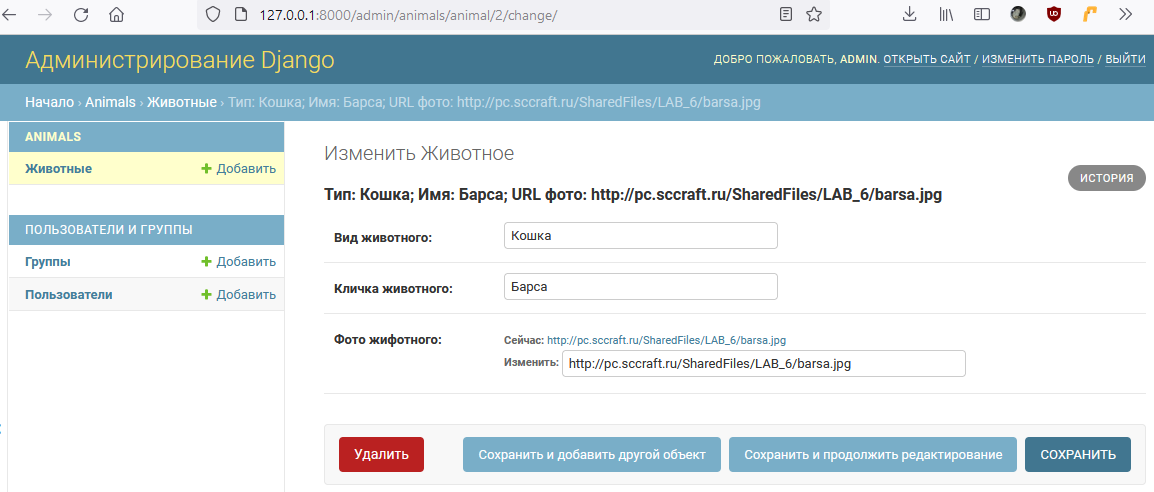


**Администрирование (/admin)**









**Обращение к API из Insomnia**

