Защищено: Гапанюк Ю.Е.		Демонстрация: Гапанюк Ю.Е.	
""	_2021 г.	""	2021 г.
	г по лабораторной рабо Разработка интернет-п ГУИМЦ		<b>Y</b>
Тема работы	: '' Разработка REST A REST Framev		анием Django
	12 (количество листо	в)	
	ИСПОЛНИТЕЛЬ: студент группы ИУ5Ц-73Б Фоминский Г.В.	(подпис	сь) 2021 г.

Москва, МГТУ - 2021

# СОДЕРЖАНИЕ

1. Описание задания	3
2. Текст программы	
3. Результаты выполнения программы	

# 1. Описание задания

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

# 2. 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

```
# 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
    11 11 11
    # 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%gba5i1wk! alyu(^grl(pmba-
# 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',
1
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',
            ],
        },
    },
1
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',
1
# 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'))
```

```
11 11 11
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
11 11 11
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()

# 3. Результаты выполнения программы

### Лог

```
"C:\Program Files\JetBrains\PyCharm 2021.2\bin\runnerw.exe" "C:\fgv9
\Sourse\RIP\Py\LAB 5\Scripts\python.exe" " C:\fqv9
\Sourse\RIP\Py\LAB 5/manage.py" runserver 8000
Performing system checks...
System check identified no issues (0 silenced).
January 18, 2022 - 17:19:59
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.
[18/Jan/2022 17:20:02] "GET / HTTP/1.1" 200 5546
[18/Jan/2022 17:20:02] "GET /static/rest framework/js/jquery-
3.5.1.min.js HTTP/1.1" 304 0
[18/Jan/2022 17:20:02] "GET /static/rest framework/js/ajax-form.js
HTTP/1.1" 304 0
[18/Jan/2022 17:20:02] "GET /static/rest framework/js/csrf.js HTTP/1.1"
304 0
[18/Jan/2022 17:20:02] "GET /static/rest framework/js/bootstrap.min.js
HTTP/1.1" 304 0
[18/Jan/2022 17:20:02] "GET /static/rest framework/js/prettify-min.js
HTTP/1.1" 304 0
[18/Jan/2022 17:20:02] "GET /static/rest framework/js/default.js
HTTP/1.1" 304 0
[18/Jan/2022 17:20:03] "GET /static/rest framework/css/prettify.css
HTTP/1.1" 200 817
[18/Jan/2022 17:20:03] "GET /static/rest framework/css/bootstrap-
tweaks.css HTTP/1.1" 200 3385
[18/Jan/2022 17:20:03] "GET /static/rest framework/css/default.css
HTTP/1.1" 200 1152
[18/Jan/2022 17:20:03] "GET /static/rest framework/css/bootstrap.min.css
HTTP/1.1" 200 121457
```



# API животных (/animals)

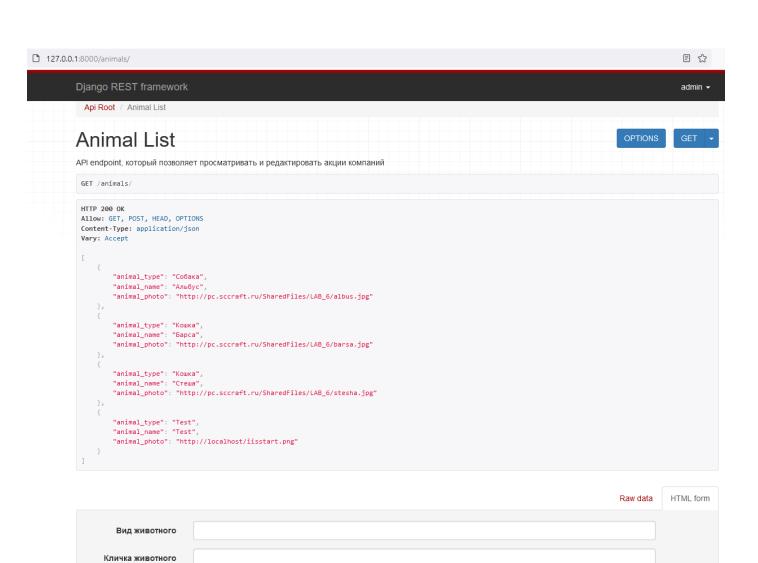
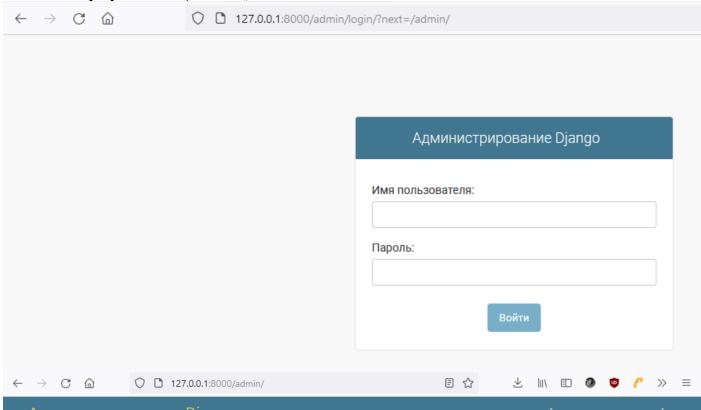


Фото жифотного

# Администрирование (/admin)



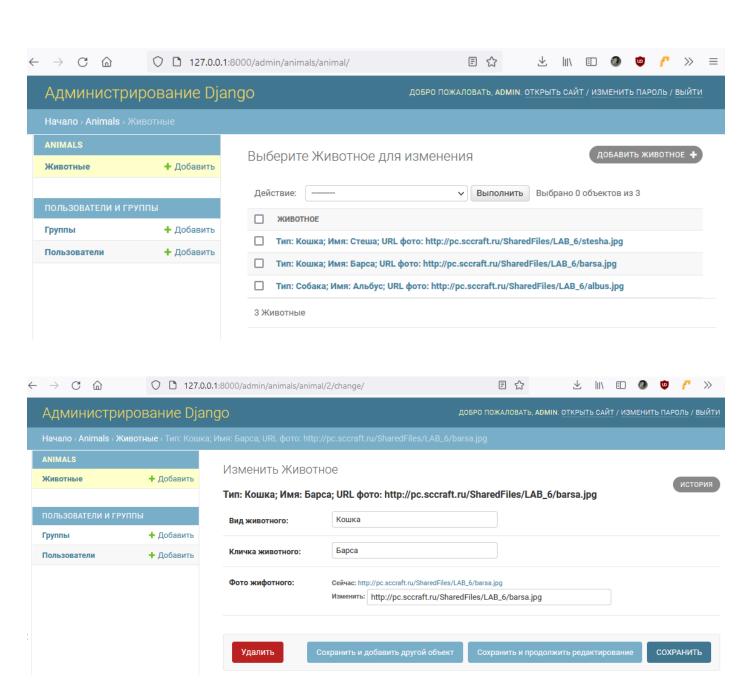
### Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ADMIN. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

### Администрирование сайта



# Мои действия Х Тип: Test; Имя: Test; URL фото: http://localhost/iisstart.png Животное Х Тип: Test; Имя: Test; URL фото: http://localhost/iisstart.png Животное Тип: Кошка; Имя: Барса; URL фото: http://pc.sccraft.ru/SharedFiles /LAB\_6/barsa.jpg Животное Тип: Кошка; Имя: Стеща; URL фото: http://pc.sccraft.ru /SharedFiles/LAB\_6/stesha.jpg Животное Тип: Кошка; Имя: Барса; URL фото: https://pc.sccraft.ru/SharedFiles /LAB\_6/barsa.jpg Животное Тип: Собака; Имя: Альбус; URL фото: http://pc.sccraft.ru /SharedFiles/LAB\_6/albus.jpg Животное



Обращение к API из Insomnia

