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

Факультет «Информатика и системы управления»

Отчёт по лабораторной работе №6	
по курсу «Разработка интернет-приложений»	<b>&gt;&gt;</b>

Разработка REST API с использованием Django REST Framework.

Выполнил:

студент группы ИУ5-51Б Абдрашитова А.Н.

Проверил:

преподаватель каф. ИУ5 Гапанюк Ю. Е.

Подпись и дата: 28.12.2021

Подпись и дата: 28.12.2021

#### Описание залания.

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

## Ход выполнения лабораторной работы

Создание модели:

### Models.py:

```
from django.db import models
# Create your models here.
# This is an auto-generated Django model module.
# You'll have to do the following manually to clean this up:
  * Rearrange models' order
  * Make sure each model has one field with primary key=True
  * Make sure each ForeignKey and OneToOneField has `on delete` set to the
desired behavior
  * Remove `managed = False` lines if you wish to allow Django to create,
modify, and delete the table
# Feel free to rename the models, but don't rename db table values or field
class Product(models.Model):
   name = models.CharField(max_length=255, blank=True, null=True)
   price = models.IntegerField(blank=True, null=True)
   quantity = models.IntegerField(blank=True, null=True)
   manufacturer = models.CharField(max length=255, blank=True, null=True)
   weight = models.FloatField(blank=True, null=True)
   expiration_date = models.DateTimeField(blank=True, null=True)
   description = models.CharField(max length=255, blank=True, null=True)
   imgSrc = models.CharField(max length=1000, blank=True, null=True)
   class Meta:
       managed = False
       db_table = 'product'
```

#### Сериализатор:

#### serializers.py:

```
from lavka.models import Product
from rest_framework import serializers

class ProductSerializer(serializers.ModelSerializer):
    class Meta:
    # Модель, которую мы сериализуем
```

```
model = Product

# Поля, которые мы сериализуем
fields = ["pk", "name", "price", "quantity", "manufacturer", "weight",
"expiration_date", "description", "imgSrc"]
```

#### views.py:

```
from django.http import HttpResponse
from django.shortcuts import render
from datetime import date
from .models import Product
from rest_framework import viewsets
from lavka.serializers import ProductSerializer
from lavka.models import Product

class ProductViewSet(viewsets.ModelViewSet):
    """

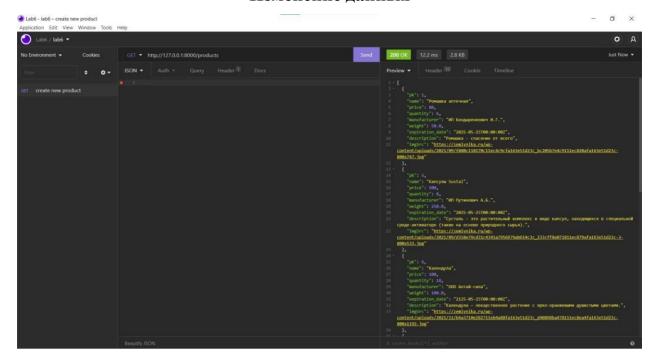
    API endpoint, который позволяет просматривать и редактировать акции компаний
    """

    # queryset всех пользователей для фильтрации по дате последнего изменения
    queryset = Product.objects.all().order_by('pk')
    serializer_class = ProductSerializer # Сериализатор для модели
```

## urls.py:

```
from django.contrib import admin
from django.urls import path
from lavka import views
from django.contrib import admin
from lavka import views as product_views
from django.urls import include, path
from rest_framework import routers
router = routers.DefaultRouter()
router.register(r'products', product views.ProductViewSet)
# 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),
```

## Изменение данных



## Изменим цену у ромашки:

