**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка Интернет-Приложений»

Отчет по лабораторной работе №6

«Разработка REST API с использованием Django REST Framework»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-53 |  | Юрий Евгеньевич Гапанюк |
| Миронова Александра |  |  |
| Подпись и дата:  24.12.21 |  | Подпись и дата: |

Москва, 2021 г.

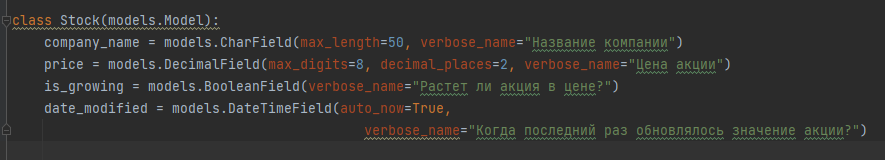
**Цель лабораторной работы:** изучение возможностей разработки REST API с использованием Django REST Framework.

**Задание:**

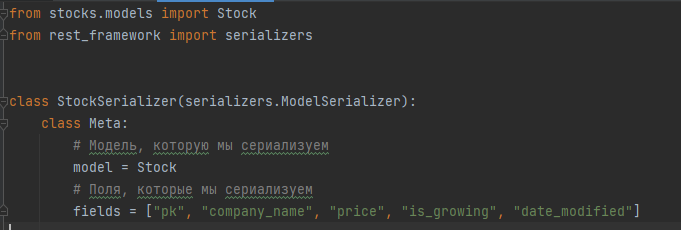
С использованием Django REST Framework разработайте REST API для одной модели (одной таблицы базы данных).

**Подготовка к выполнению программы:**

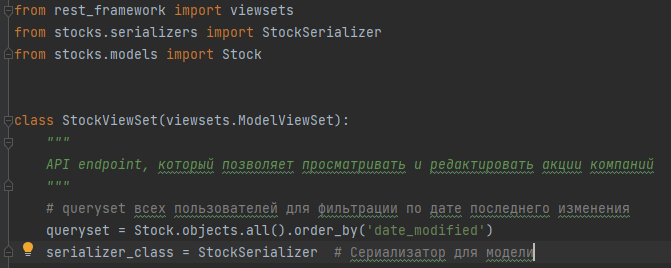
Мы создали модель акции в коде (class Stock), и затем, создали такую таблицу при помощи миграции.



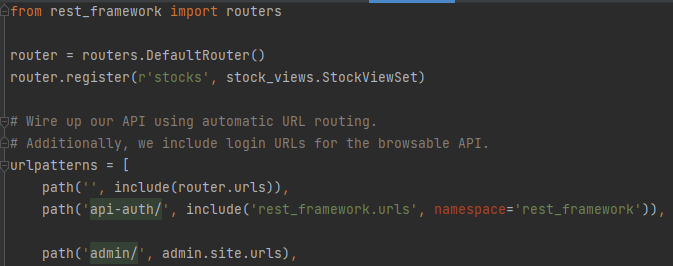
Затем был создан сериализатор, чтобы преобразовывать нашу модель из базы данных в JSON

****

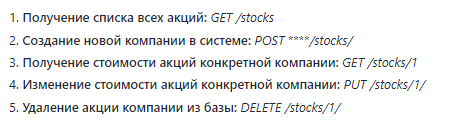
Далее в view создан запрос, который будет посылать запросы в базу данных



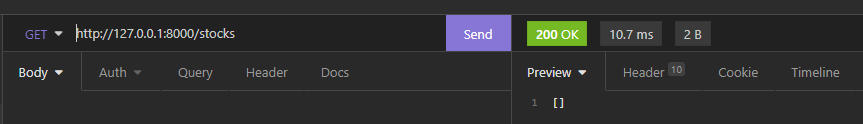
Добавили роутер нашего view в URL'ы приложения



Виды исполняемых запросов

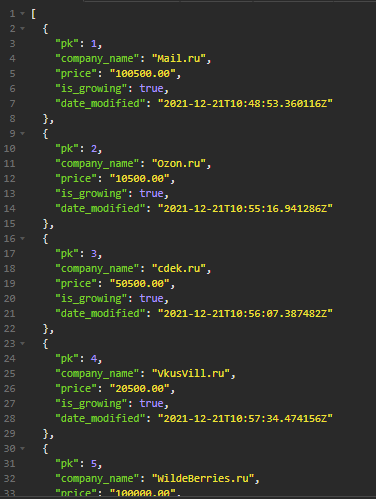


С помощью программы Insomnia был отправлен запрос на вывод всех акций и выведен его результат. Получен пустой список, тк база данных акций пока пуста

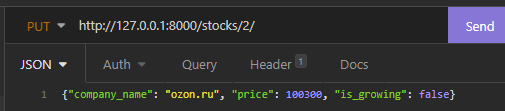


Поэтому деле мы добавляем в список акции новой компании при помощи HTTP метода POST

{"company\_name": "WildeBerries.ru", "price": 100000, "is\_growing": false}



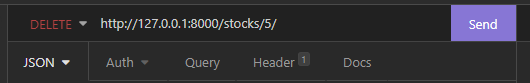
Теперь мы попробовали изменить информацию об акции с ключевым номером 2



Проверили, что в общем списке акций ее данные изменены



Далее мы удалили объект с названием WildeBerries.ru. Это запись с pk=5.





**Выполнение задания лабораторной работы**

**Текст программы:**

**views.py**

from rest\_framework import viewsets

from stocks.serializers import StockSerializer

from stocks.models import IceCreamStock

class StockViewSet(viewsets.ModelViewSet):

"""

API endpoint, который позволяет просматривать и редактировать акции компаний

"""

# queryset всех пользователей для фильтрации по primary key

queryset = IceCreamStock.objects.all().order\_by('pk')

serializer\_class = StockSerializer # Сериализатор для модели

**models.py**

from django.db import models

class IceCreamStock(models.Model):

title = models.CharField(max\_length=30, verbose\_name="Название мороженного")

ingridients = models.CharField(max\_length=200, verbose\_name="Ингридиенты в составе мороженного")

standart = models.CharField(max\_length=30, verbose\_name="Стандарт мороженного")

price = models.DecimalField(max\_digits=8, decimal\_places=2, verbose\_name="Цена мороженного")

date\_modified = models.DateTimeField(auto\_now=True,

verbose\_name="Когда последний раз обновлялась информация?")

**serializers.py**

from stocks.models import IceCreamStock

from rest\_framework import serializers

class StockSerializer(serializers.ModelSerializer):

class Meta:

# Модель, которую мы сериализуем

model = IceCreamStock

# Поля, которые мы сериализуем

fields = ["pk", "title", "ingridients", "standart", "price", "date\_modified"]

**urls.py**

from django.contrib import admin

from stocks import views as stock\_views

from django.urls import include, path

from rest\_framework import routers

router = routers.DefaultRouter()

router.register(r'stocks', stock\_views.StockViewSet)

# Wire up our API using automatic URL routing.

# Additionally, we include login URLs for the browsable API.

urlpatterns = [

path('', include(router.urls)),

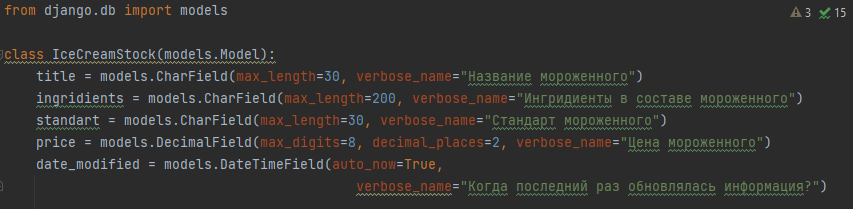
path('api-auth/', include('rest\_framework.urls', namespace='rest\_framework')),

path('admin/', admin.site.urls),

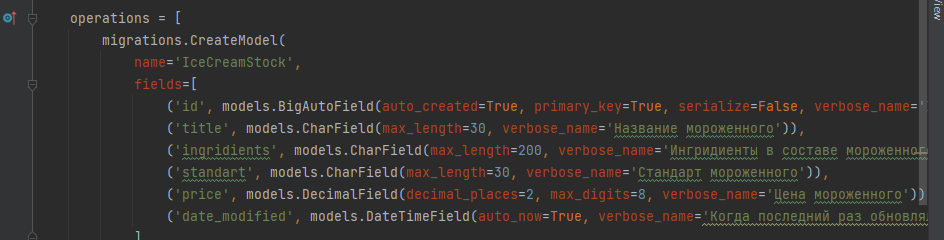
]

**Экранные формы с примерами выполнения программы:**

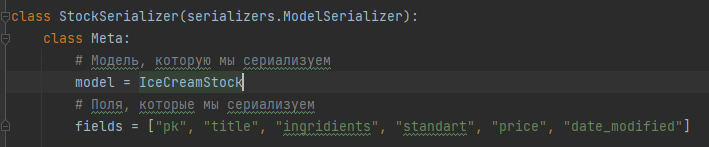
Была создана модель, содержащая информацию о мороженом (class IcecreamStock). Затем, при помощи миграции Создана соответствующая таблица.

****

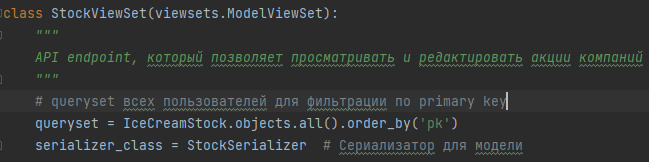
Файл миграций

****

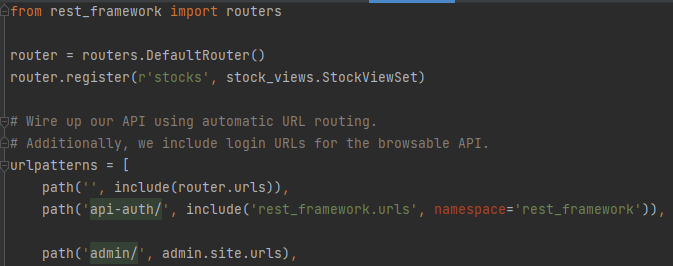
Затем был создан сериализатор, чтобы преобразовывать нашу модель из базы данных в JSON

****

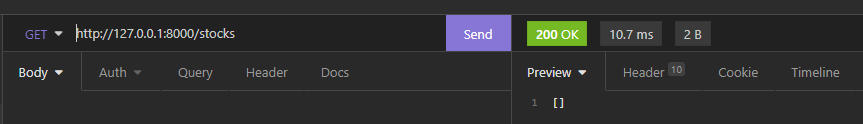
Далее в view создан запрос, который будет посылать запросы в базу данных



Добавили роутер нашего view в URL'ы приложения



С помощью программы Insomnia был отправлен запрос на вывод всех акций и выведен его результат. Получен пустой список, тк база данных пока пуста



Список видов мороженного был заполнен при помощи HTTP метода POST

{"title": "Пломбир", "ingridients": "Сливки 33%, Молоко, Яичный желток, Сахарная пудра, Ванилин",

"standart": "ГОСТ 119-41", "price": 72}

{"title": "Шоколадное мороженое", "ingridients": "Сливки 33%, Молоко, Яичный желток, Сахарная пудра, Какао",

"standart": "ГОСТ 31457-2012", "price": 68}

{"title": "Клубничное мороженое", "ingridients": "Сливки 33%, Молоко, Яичный желток, Сахарная пудра, Клубника",

"standart": "ГОСТ 31457-2012", "price": 62}

{"title": "Фисташковое мороженое", "ingridients": "Сливки 33%, Молоко, Яичный желток, Сахарная пудра, Фисташки",

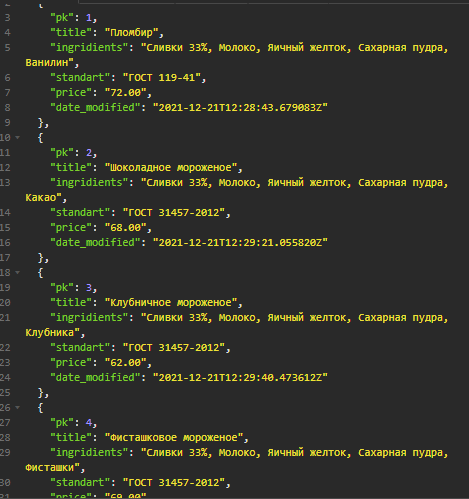
"standart": "ГОСТ 31457-2012", "price": 69}

{"title": "Крем-брюле",

"ingridients": "Сливки 33%, Молоко, Яичный желток, Сахарная пудра, Сгущеное молоко, Ванилин, Крахмал кукурузный",

"standart": "ГОСТ 31457-2012", "price": 81}

**Результат заполнения**

****

Аналогично действиям, проделанным в подготовке к лаб.работе можно изменять, удалять записи в базе данных при помощи методов PUT и DELETE. А так же запрашивать информацию конкретной записи по ее параметру методом GET.