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 рубежному контролю №2
Вариант В-6

Выполнил:
студент группы ИУ5-53Б
Ветошкин А.А,

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Ветошкин Артём Алексеевич ИУ5-53Б

Задание:

Вариант В-6.

Есть сущности "Процессор" и "Комьютер". *Требуется сделать*:

- Создайте проект Python Django с использованием стандартных средств Django.
- Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
- С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
- Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

Создание моделей и "ручек" для работы с ними

Создание моделей:

```
class PCManager(models.Manager):
    def get_pc_by_processor_id(self, id):
        return self.filter(id_processor=id).all()


class Processor(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=30, blank=True, null=False)
    frequency = models.IntegerField()
    memory_cash = models.IntegerField()

    class Meta:
        db_table = 'processor'


class PC(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=30, blank=True, null=False)
    price = models.IntegerField()
    id_processor = models.ForeignKey('Processor', models.DO_NOTHING,
db_column='id_processor', blank=True, null=True)

    objects = PCManager()

    class Meta:
```

```
        db_table = 'pc'
```

Для возможности добавления, редактирования и удаления данных были использован rest_framework.

Сотвественно был прописаны сериализаторы:

```
class PCSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.PC
        fields = ["id", "name", "price", "id_processor"]


class ProcessorSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.Processor
        fields = ["id", "name", "frequency", "memory_cash"]
```

И представления:

```
class PCViewSet(viewsets.ModelViewSet):
    queryset = models.PC.objects.all()
    serializer_class = serializers.PCSerializer


class ProcessorViewSet(viewsets.ModelViewSet):
    queryset = models.Processor.objects.all()
    serializer_class = serializers.ProcessorSerializer
```

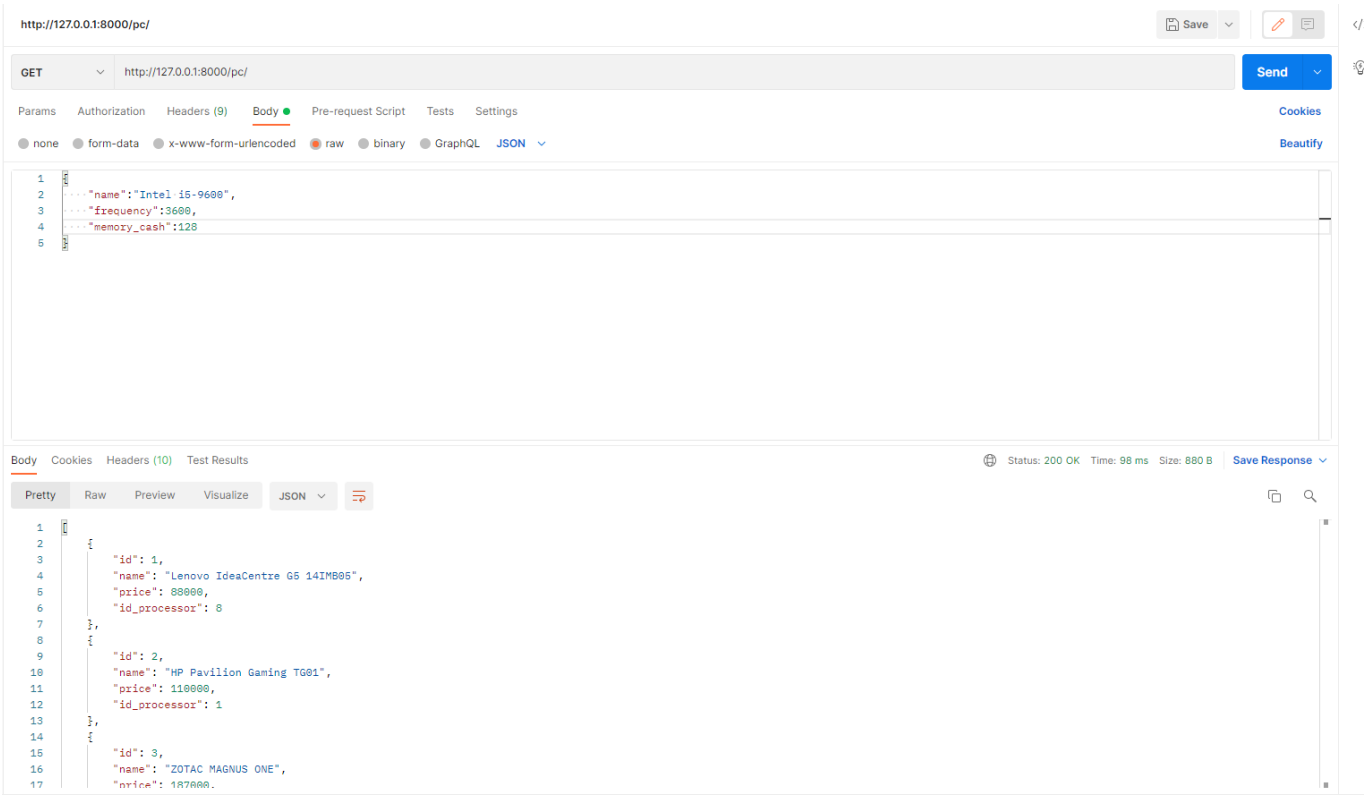
И настроен роутер:

```
router = routers.DefaultRouter()
router.register('processor', views.ProcessorViewSet)
router.register('pc', views.PCViewSet)
```

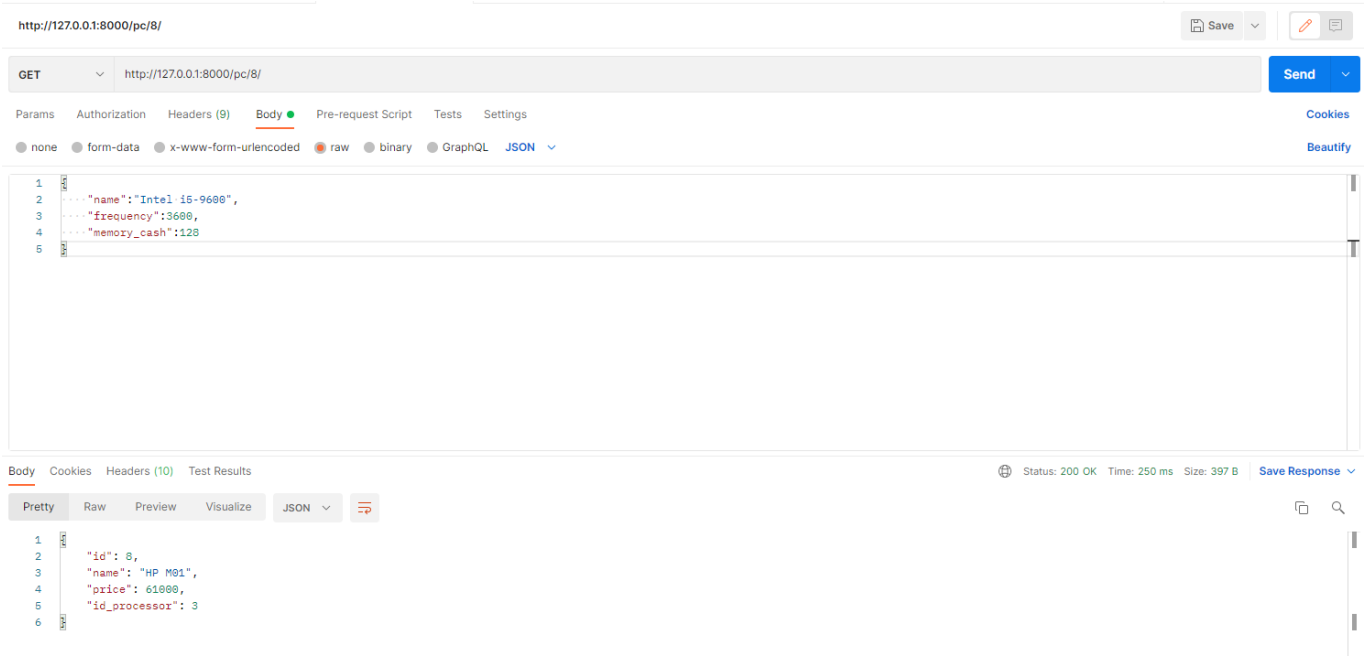
Проверка выполнения запросов на созданные ручик

Для записей о компьютерах

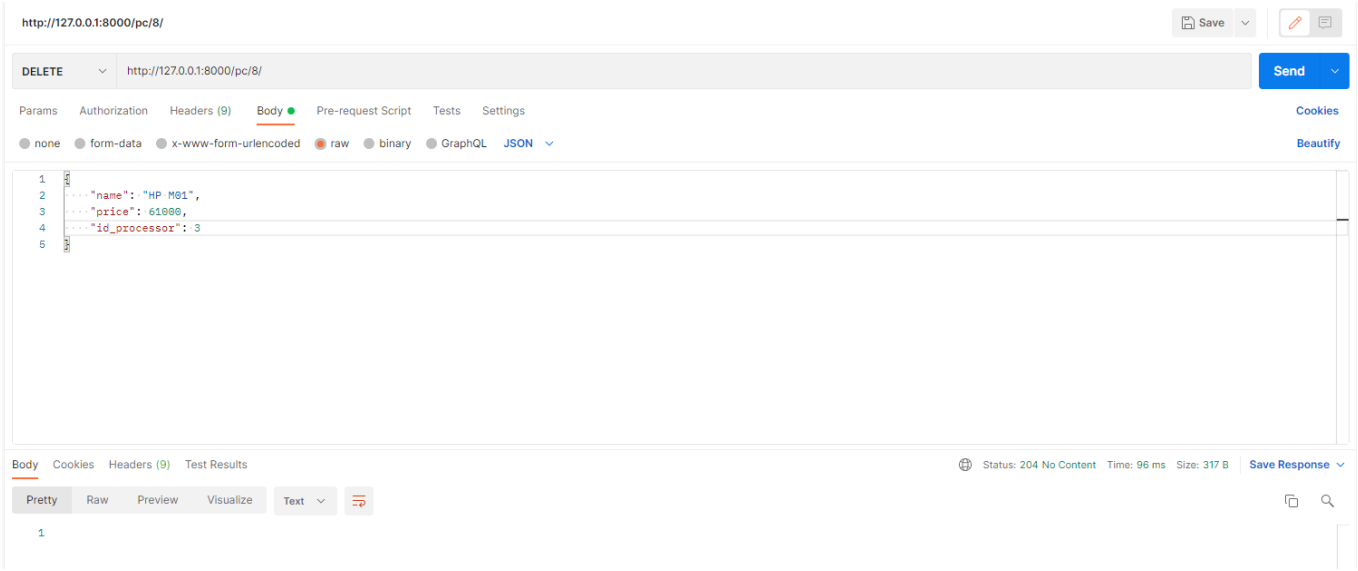
Получение всех компьютеров GET запросом:



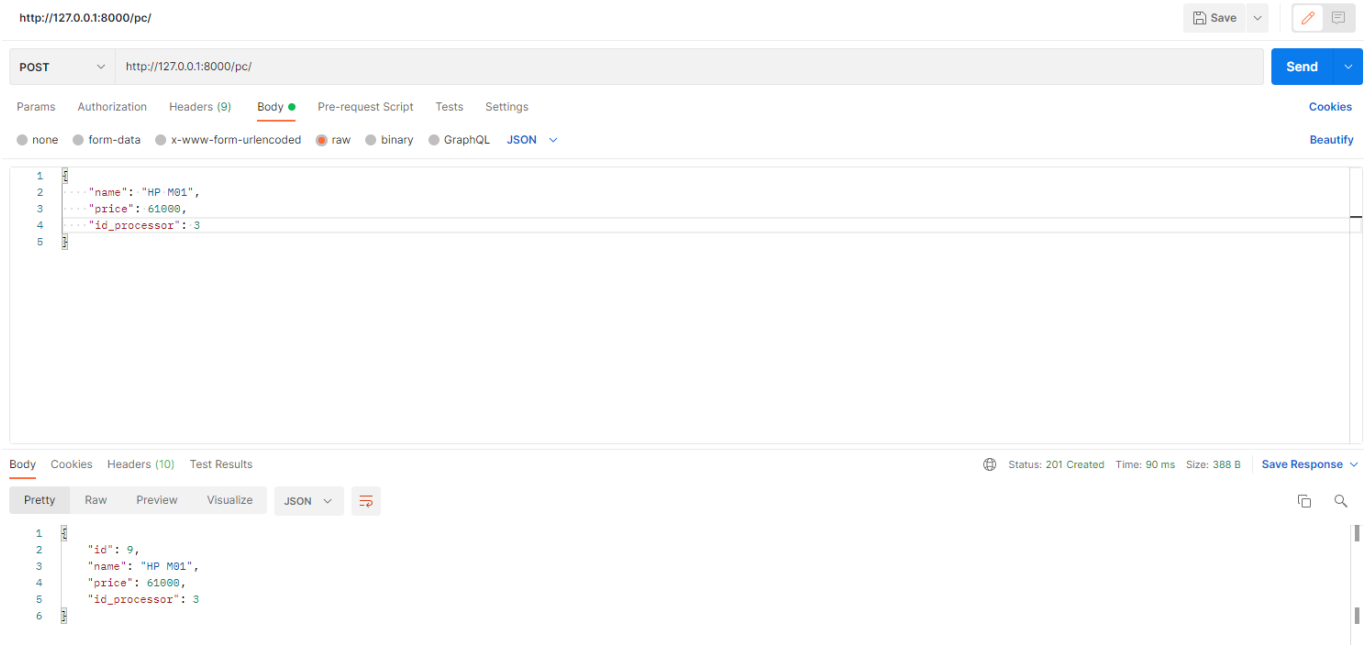
Получение конкретного компьютера GET запросом:



Удаление конкретного компьютера DELETE запросом:

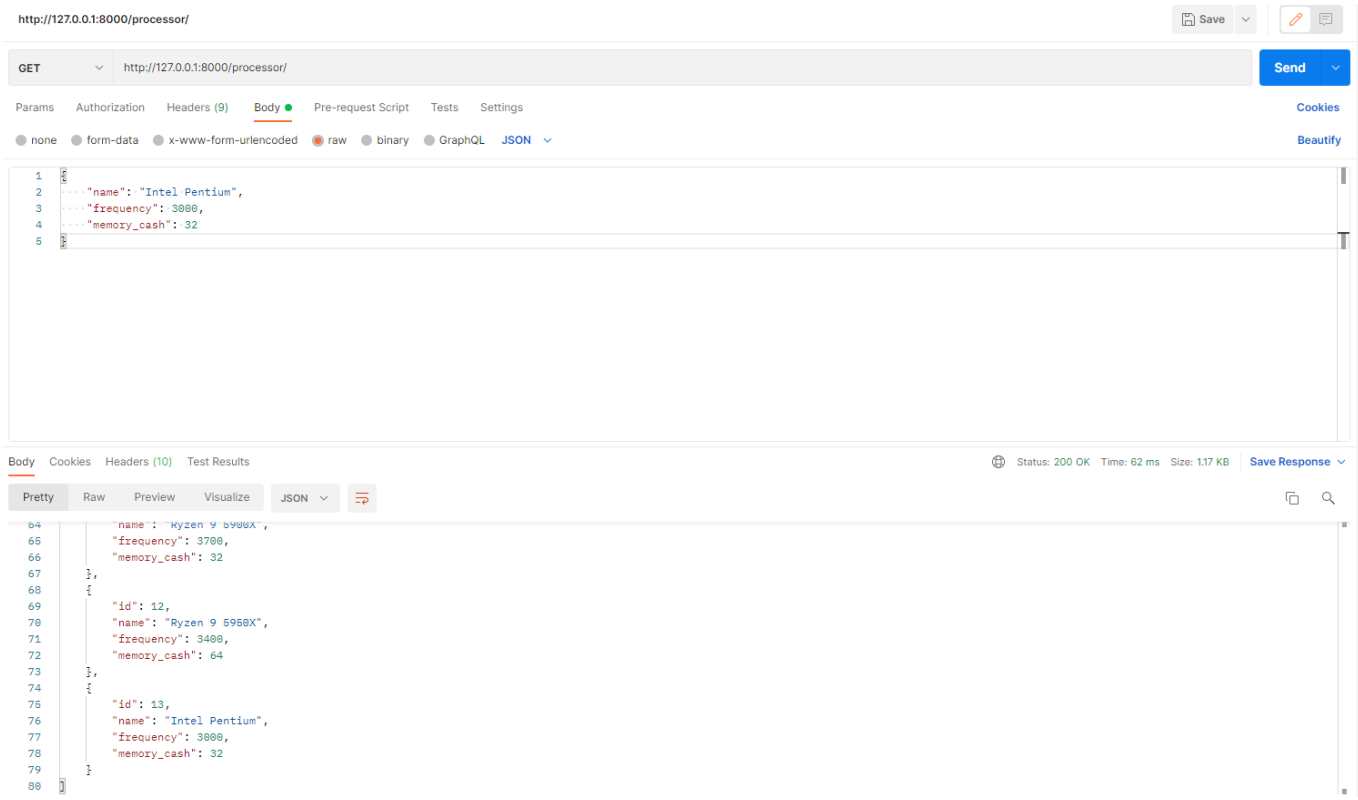


Создание записи о компьютере POST запросом:

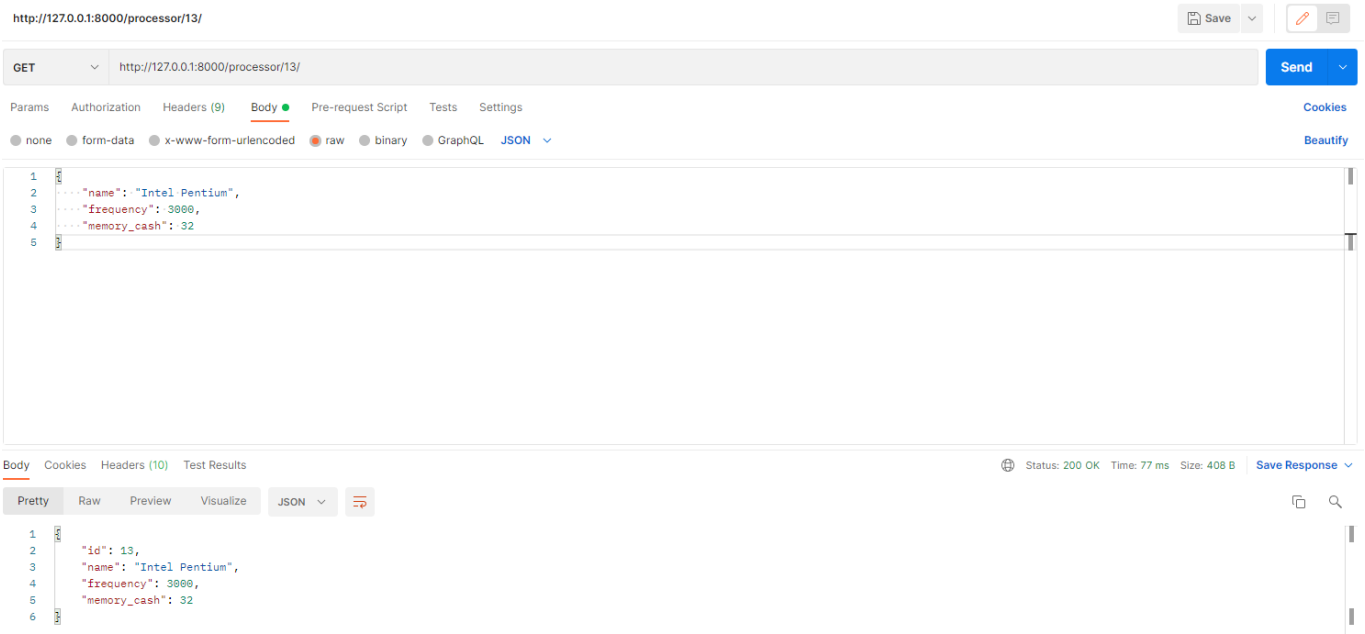


Для записей о процессорах

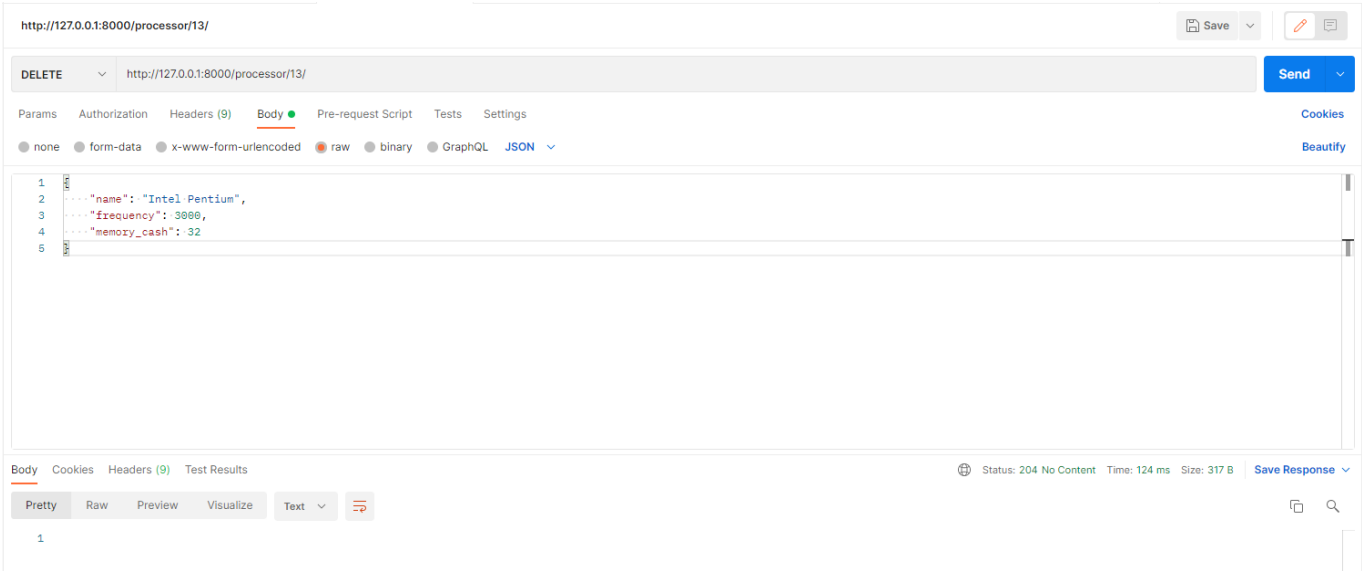
Получение всех процессоров GET запросом:



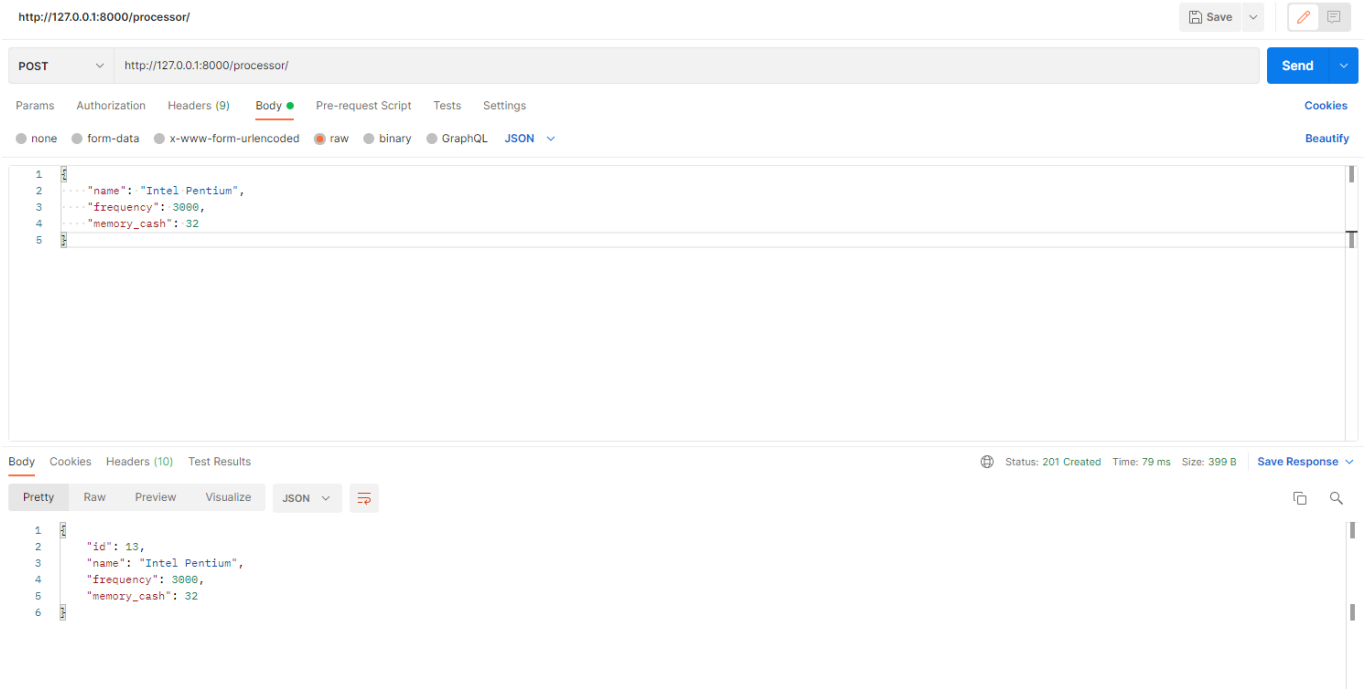
Получение конкретного процессора GET запросом:



Удаление конкретного процессора DELETE запросом:



Создание записи о процессоре POST запросом:



Создание отчёта

Для показа отчёта было добавленно представление:

```
def index(request):
    processors = models.Processor.objects.all()
    data = []
    for processor in processors:
        data.append((processor,
models.PC.objects.get_pc_by_processor_id(processor.id)))

    return render(request, "index.html", {"data": data})
```

И сверстана страница:

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Report</title>
    <style>
        table.collapse {
            border: 2px solid black;
            border-collapse: collapse;
            margin-top: 10px;
            display: inline-block;
        }

        td.upper {
            border: 2px solid black;
            padding: 10px;
        }

        td {
            border: 2px solid black;
            padding: 5px;
        }

        .center-text {
            text-align: center;
        }
    </style>
</head>
<body>
<h1 class="center-text"> Отчёт по данным базы о процессорах и компьютерах</h1>
<div class="center-text">
    {% for value in data %}
        <table class="collapse center-text">
            <tr>
                <td class="upper" colspan="12">Процессор</td>
            </tr>
            <tr>
                <td class="upper" colspan="3">ID</td>
                <td class="upper" colspan="3">Название</td>
                <td class="upper" colspan="3">Частота</td>
                <td class="upper" colspan="3">Размер кэша</td>
            </tr>
            <tr>
                <td class="upper" colspan="3">{{ value.0.id }}</td>
                <td class="upper" colspan="3">{{ value.0.name }}</td>
                <td class="upper" colspan="3">{{ value.0.frequency }}</td>
                <td class="upper" colspan="3">{{ value.0.memory_cash }}</td>
            </tr>
            <tr>
                <td class="upper" colspan="12">Используется в компьютерах</td>
            </tr>
            <tr>
                <td colspan="4">ID</td>
```

```
                <td colspan="4">Название</td>
                <td colspan="4">Цена</td>
            </tr>
            {% for pc in value.1 %}
                <tr>
                    <td colspan="4">{{ pc.id }}</td>
                    <td colspan="4">{{ pc.name }}</td>
                    <td colspan="4">{{ pc.price }}</td>
                </tr>
            {% empty %}
                <tr>
                    <td colspan="12">Не найдено</td>
                </tr>
            {% endfor %}
        </table>
    {% endfor %}
</div>
</body>
</html>
```

Для заполнения данных был написан sql скрипт:

```
INSERT INTO processor (name, frequency, memory_cash) VALUES
    ('Intel i7-10600', 3700, 128),
    ('Intel i5-6500', 3600, 12),
    ('Intel i5-6600', 3700, 32),
    ('Intel i7-8600', 3500, 64),
    ('Intel i7-10600', 3800, 256),
    ('Intel i3-5600', 3500, 128),
    ('Intel i7-4790', 3200, 128),
    ('Ryzen 5 5600X', 3800, 32),
    ('Ryzen 7 5800X', 3700, 32),
    ('Ryzen 9 5900X', 3700, 32),
    ('Ryzen 9 5950X', 3400, 64);

INSERT INTO PC (name, price, id_processor) VALUES
    ('Lenovo IdeaCentre G5 14IMB05', 88000, 8),
    ('HP Pavilion Gaming TG01', 110000, 1),
    ('ZOTAC MAGNUS ONE', 187000, 7),
    ('Apple Mac mini 2020 M1', 173000, 6),
    ('Acer Aspire TC-895', 34000, 7),
    ('Apple iMac 24 2021', 248000, 11),
    ('Gigabyte GB-BR', 93000, 4),
    ('HP M01', 61000, 3);
```

В итоге получился отчёт следующего вида:

Отчёт по данным базы о процессорах и компьютерах

Процессор			
ID	Название	Частота	Размер кэша
1	Intel i5-9600	3600	128
Используется в компьютерах			
ID	Название	Цена	
2	HP Pavilion Gaming TG01	110000	

Процессор			
ID	Название	Частота	Размер кэша
2	Intel i7-10600	3700	128
Используется в компьютерах			
ID	Название	Цена	
Не найдено			

Процессор			
ID	Название	Частота	Размер кэша
3	Intel i5-6500	3600	12
Используется в компьютерах			
ID	Название	Цена	
9	HP M01	61000	

Процессор			
ID	Название	Частота	Размер кэша
4	Intel i5-6600	3700	32
Используется в компьютерах			
ID	Название	Цена	
7	Gigabyte GB-BR	93000	

Процессор			
ID	Название	Частота	Размер кэша
5	Intel i7-8600	3500	64
Используется в компьютерах			
ID	Название	Цена	
Не найдено			

Процессор			
ID	Название	Частота	Размер кэша
6	Intel i7-10600	3800	256
Используется в компьютерах			
ID	Название	Цена	
4	Apple Mac mini 2020 M1	173000	

Процессор			
ID	Название	Частота	Размер кэша
7	Intel i3-5600	3500	128
Используется в компьютерах			
ID	Название	Цена	
3	ZOTAC MAGNUS ONE	187000	
5	Acer Aspire TC-895	34000	

Процессор			
ID	Название	Частота	Размер кэша
8	Intel i7-4790	3200	128
Используется в компьютерах			
ID	Название	Цена	
1	Lenovo IdeaCentre G5 14IMB05	88000	

Процессор			
ID	Название	Частота	Размер кэша
9	Ryzen 5 5600X	3800	32
Используется в компьютерах			
ID	Название	Цена	
Не найдено			

Процессор			
ID	Название	Частота	Размер кэша
10	Ryzen 7 5800X	3700	32
Используется в компьютерах			
ID	Название	Цена	
Не найдено			

Процессор			
ID	Название	Частота	Размер кэша
11	Ryzen 9 5900X	3700	32
Используется в компьютерах			
ID	Название	Цена	
6	Apple iMac 24 2021	248000	

Процессор			
ID	Название	Частота	Размер кэша
12	Ryzen 9 5950X	3400	64
Используется в компьютерах			
ID	Название	Цена	
Не найдено			