



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

Рубежный контроль № 2
по курсу “Разработка интернет-приложений”

Выполнил:
студент группы ИУ5-53Б
Волгина А. Д.
17.12.21

Проверил:
Гапанюк Ю.Е.

2021 г.

Для сущностей “Компьютер” и “Жёсткий диск” сделать:

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

Создадим модель:

```
from django.db import models

class DiskManager(models.Manager):
    def get_disk_by_comp_id(self, id):
        return self.filter(computerID=id).all()

class Computer(models.Model):
    company = models.CharField(max_length=30)
    price = models.IntegerField()

class HardDrive(models.Model):
    computerID = models.ForeignKey('Computer', models.DO_NOTHING, db_column='computerID', blank=True, null=True)
    name = models.CharField(max_length=30)
    capacity = models.IntegerField()
    filled = models.IntegerField()

objects = DiskManager()
```

Для возможности добавления, редактирования и удаления данных был использован `rest_framework`.

Сериализаторы:

```
from django.db import models

class DiskManager(models.Manager):
    def get_disk_by_comp_id(self, id):
        return self.filter(computerID=id).all()

class Computer(models.Model):
    company = models.CharField(max_length=30)
    price = models.IntegerField()

class HardDrive(models.Model):
    computerID = models.ForeignKey('Computer', models.DO_NOTHING, db_column='computerID', blank=True, null=True)
    name = models.CharField(max_length=30)
    capacity = models.IntegerField()
    filled = models.IntegerField()

objects = DiskManager()
```

Представления:

```

from rest_framework import viewsets
from rk2_app.serializers import ComputerSerializer, HardDriveSerializer
from rk2_app.models import Computer, HardDrive

class ComputerViewSet(viewsets.ModelViewSet):
    """
    API endpoint, который позволяет просматривать и редактировать акции компаний
    """
    queryset = Computer.objects.all()
    serializer_class = ComputerSerializer

class HardDriveViewSet(viewsets.ModelViewSet):
    """
    API endpoint, который позволяет просматривать и редактировать акции компаний
    """
    queryset = HardDrive.objects.all()
    serializer_class = HardDriveSerializer

```

Роутер:

```

from rk2_app import views
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'computers', views.ComputerViewSet)
router.register(r'hard_drives', views.HardDriveViewSet)

urlpatterns = [
    path('', include(router.urls)),
    path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),
    path('admin/', admin.site.urls),
]

```

Макет веб-приложения.

Добавим данные в наши таблицы:

Компьютеры

POST http://127.0.0.1:8000/computers/

JSON	Auth	Query 1	Header 1
1 {			
2 "company": "Intel",			
3 "price": 3000			
4 }			
5			

```
POST http://127.0.0.1:8000/computers/

JSON
Auth Query Headers

1 {
2   "company": "HP",
3   "price": 3500
4 }
```

Посмотрим получившуюся таблицу:

```
GET http://127.0.0.1:8000/computers/

1 [
2   {
3     "pk": 5,
4     "company": "Intel",
5     "price": 3000
6   },
7   {
8     "pk": 6,
9     "company": "Intel",
10    "price": 3000
11  },
12  {
13    "pk": 7,
14    "company": "HP",
15    "price": 3500
16  }
17 ]
```

Отредактируем 6 компьютер:

```
PUT http://127.0.0.1:8000/computers/6/

JSON
Auth Query Headers

1 {
2   "company": "Dell",
3   "price": 5000
4 }
```

```

1 ▾ [
2 ▾ {
3     "pk": 5,
4     "company": "Intel",
5     "price": 3000
6 },
7 ▾ {
8     "pk": 6,
9     "company": "Dell",
10    "price": 5000
11 },
12 ▾ {
13     "pk": 7,
14     "company": "HP",
15     "price": 3500
16 }
17 ]

```

Жёсткие диски

```

1 ▾ [
2 ▾ {
3     "pk": 1,
4     "computerID": 5,
5     "name": "C",
6     "capacity": 300,
7     "filled": 200
8 },
9 ▾ {
10    "pk": 2,
11    "computerID": 7,
12    "name": "C",
13    "capacity": 300,
14    "filled": 100
15 },
16 ▾ {
17    "pk": 3,
18    "computerID": 6,
19    "name": "C",
20    "capacity": 200,
21    "filled": 150
22 },
23 ▾ {
24    "pk": 4,
25    "computerID": 5,
26    "name": "E",
27    "capacity": 200,
28    "filled": 50
29 },
30 ▾ {
31    "pk": 5,
32    "computerID": 5,
33    "name": "F",
34    "capacity": 100,
35    "filled": 0
36 },
37 ▾ {
38    "pk": 6,
39    "computerID": 5,
40    "name": "L",
41    "capacity": 100,
42    "filled": 0

```

```

42     "filled": 0
43   },
44   {
45     "pk": 7,
46     "computerID": 6,
47     "name": "E",
48     "capacity": 150,
49     "filled": 0
50   },
51   {
52     "pk": 8,
53     "computerID": 6,
54     "name": "F",
55     "capacity": 250,
56     "filled": 50
57   },
58   {
59     "pk": 9,
60     "computerID": 7,
61     "name": "F",
62     "capacity": 250,
63     "filled": 0
64   }
65 ]

```

Удалим 9 диск:

DELETE http://127.0.0.1:8000/hard_drives/9/

```

44   {
45     "pk": 7,
46     "computerID": 6,
47     "name": "E",
48     "capacity": 150,
49     "filled": 0
50   },
51   {
52     "pk": 8,
53     "computerID": 6,
54     "name": "F",
55     "capacity": 250,
56     "filled": 50
57   }
58 ]

```

А теперь сформируем отчёт.

Для этого создадим шаблон:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Report</title>
</head>
<body>
<h1 align="center">Отчёт по данным о компьютерах и дисках</h1>
<div align="center" class="text-center">
  <table border="1">
    <tr>
      <td>ID компьютера</td>
      <td>Компания</td>
      <td>Цена</td>
      <td>
        <table align="center">
          <tr align="center">
            <td>ID диска</td>
            <td>Название</td>
            <td>Объём</td>
            <td>Занято</td>
          </tr>
        </table>
      </td>
    </tr>
    {% for value in data %}
      <tr>
        <td>{{ value.0.pk }}</td>
        <td>{{ value.0.company }}</td>

```

```

{% for value in data %}
    <tr>
        <td>{{ value.0.pk }}</td>
        <td>{{ value.0.company }}</td>
        <td>{{ value.0.price }}</td>
        <td>
            <table align="center" width="100%">
                {% for disk in value.1 %}
                    <tr align="center">
                        <td width="25%">{{ disk.pk }}</td>
                        <td width="25%">{{ disk.name }}</td>
                        <td width="25%">{{ disk.capacity }}</td>
                        <td width="25%">{{ disk.filled }}</td>
                    </tr>
                {% endfor %}
            </table>
        </td>
    </tr>
{% endfor %}
</table>
</div>
</body>
</html>

```

И представление:

```

def report(request):
    computers = Computer.objects.all()
    data = []
    for comp in computers:
        data.append((comp, HardDrive.objects.get_disk_by_comp_id(comp.id)))
    return render(request, "index.html", {"data": data})

```


Отчёт по данным о компьютерах и дисках

ID компьютера	Компания	Цена	ID диска	Название	Объём	Занято
5	Intel	3000	1	C	300	200
			4	E	200	50
			5	F	100	0
			6	L	100	0
6	Dell	5000	3	C	200	150
			7	E	150	0
			8	F	250	50
7	HP	3500	2	C	300	100