

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

Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по рубежному контролю №2

Вариант Е19

Выполнил:

студент группы РТ5-51Б
Сысоев Александр

Проверил:

доцент каф. ИУ5
Гапанюк Ю. Е.

Москва, 2021 г.

ПОЛУЧЕННОЕ ЗАДАНИЕ

Рубежный контроль представляет собой разработку веб-приложения с использованием фреймворка Django. Веб-приложение должно выполнять следующие функции:

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.

ИСХОДНЫЙ КОД

Файл models.py:

```
from django.db import models

class Manufacturers(models.Model):
    name_man = models.CharField('Имя производителя', max_length=200)

class Details(models.Model):
    name_det = models.CharField('Наименование детали', max_length=200)
    cost_det = models.IntegerField('Цена детали', default=0)
    mans = models.ForeignKey(Manufacturers, models.DO NOTHING)
```

Файл views.py:

```
from django.shortcuts import render
from rest_framework import viewsets
from Factory.models import Manufacturers, Details
from Factory.serializers import ManSerializer, DetSerializer

def index(request):
    return render(request, 'index.html')

class ManViewSet(viewsets.ModelViewSet):
    queryset = Manufacturers.objects.all().order_by("name_man")
    serializer_class = ManSerializer
```

```

class DetViewSet(viewsets.ModelViewSet):
    queryset = Details.objects.all().order_by("name_det")
    serializer_class = DetSerializer

def report(request):
    return render(request, 'report.html', {'data': {'details':
Details.objects.select_related('mans')}})

```

Файл urls.py:

```

from django.contrib import admin
from django.urls import include, path
from Factory.views import ManViewSet, DetViewSet
import Factory.views
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'Manufacturer', ManViewSet)
router.register(r'Detail', DetViewSet)

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

```

Файл Factory/urls.py:

```

from django.urls import path

from . import views

urlpatterns = [
    path('', views.index),
    path('Manufacturer/', views.manufacturers),
    path('Detail/', views.details),
    path('Manufacturer/<int:id>', views.manufacturer),
    path('Program/<int:id>', views.detail),
]

```

Файл report.html:

```

{% block title %} Рубежный контроль №2. Сысоев Александр РТ5-51Б. Вариант Е19
{% endblock %}
{% block body %}
<div>
<div>
<table>
<thead>
<tr>
<th scope="col">ID</th>
<th scope="col">Наименование детали</th>
<th scope="col">Цена детали</th>
<th scope="col">Производитель</th>
</tr>
</thead>

```

```
<tbody>
{% for det in data.details %}
<tr>
<th scope="row">{{ det.id }}</th>
<td>{{ det.name_det }}</td>
<td>{{ det.cost_det }}</td>
<td>{{ det.mans.name_man }}</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
{% endblock %}
```

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

GET-запросы:

Man List

OPTIONS

GET ▾

GET /Manufacturer/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 3,
    "name_man": "Иванов"
  },
  {
    "id": 4,
    "name_man": "Петров"
  },
  {
    "id": 1,
    "name_man": "Производитель №1"
  },
  {
    "id": 2,
    "name_man": "Производитель №2"
  },
  {
    "id": 5,
    "name_man": "Сидоров"
  }
]
```

Det List

OPTIONS

GET ▾

GET /Detail/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 6,
    "name_det": "Болт",
    "cost_det": 60,
    "mans": 1
  },
  {
    "id": 4,
    "name_det": "Винт",
    "cost_det": 40,
    "mans": 4
  },
  {
    "id": 1,
    "name_det": "Гайка",
    "cost_det": 20,
    "mans": 1
  },
  {
    "id": 3,
    "name_det": "Гвоздь",
    "cost_det": 10,
    "mans": 2
  }
]
```

Det Instance

DELETE OPTIONS GET ▾

GET /Detail/1/

HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
{
  "id": 1,
  "name_det": "Гайка",
  "cost_det": 20,
  "mans": 1
}
```

Raw data HTML form

Наименование детали	<input type="text" value="Гайка"/>
Цена детали	<input type="text" value="20"/>
Mans	<input type="text" value="Manufacturers object (1)"/>

PUT

Man Instance

DELETE OPTIONS GET ▾

GET /Manufacturer/1/

HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
{
  "id": 1,
  "name_man": "Производитель №1"
}
```

Raw data HTML form

Имя производителя	<input type="text" value="Производитель №1"/>
-------------------	---

PUT

POST-запросы:

```
{
  "id": 5,
  "name_man": "Сидоров"
}
```

Raw data

HTML form

Имя производителя

Сидоров

POST

```
{
  "id": 2,
  "name_det": "Шуруп",
  "cost_det": 50,
  "mans": 2
}
```

Raw data

HTML form

Наименование детали

Шуруп

Цена детали

50

Mans

Manufacturers object (2)

POST

DELETE-запрос:

DELETE

http://127.0.0.1:8000/Detail/4/

Send

JSON

Auth

Query

Header 1

Docs

1

...

PUT-запрос:

PUT

http://127.0.0.1:8000/Detail/4/

Send

200 OK

246 ms

59 B

JSON

Auth

Query

Header 1

Docs

Preview

Header 10

Cookie

Timeline

1 {
2 "name_det": "Саморез",
3 "mans": 4
4 }

1 {
2 "id": 4,
3 "name_det": "Саморез",
4 "cost_det": 40,
5 "mans": 4
6 }

Отчёт, соединяющий данные из 2-х таблиц:

Рубежный контроль №2. Сысоев Александр РТ5-51Б. Вариант Е19

ID	Наименование детали	Цена детали	Производитель
1	Гайка	20	Производитель №1
2	Шуруп	50	Производитель №2
3	Гвоздь	10	Иванов
4	Саморез	40	Петров
5	Штуцер	100	Сидоров
6	Болт	60	Производитель №1