

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-15

Выполнил:
студент группы ИУ5-53Б
Ларионова А.П.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

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.

Класс1-Файл

Класс2-Каталог файлов

Описание предметной области:

Классы данных для предметной области Файл-Каталог файлов:

1. Класс «Файл», содержащий поля:
 - ID файла
 - название файла
 - размер файла
 - расширение файла
 - ID каталога файлов (для реализации связи 1-М)
2. Класс «Каталог файлов», содержащий поля:
 - ID каталога файлов
 - название каталога

1)С помощью Django ORM создаем модели:

```
models.py x 0001_initial.py x settings.py x serializers.py x views.py x urls.py x
1 from django.db import models
2
3 class File(models.Model):
4     id_file = models.AutoField('id_file', primary_key=True)
5     name = models.CharField(max_length=40, verbose_name="Название")
6     size = models.IntegerField(verbose_name="Размер")
7     extension = models.CharField(max_length=20, verbose_name="Расширение")
8     id_catalog = models.ForeignKey('Catalog', models.DO_NOTHING, db_column='id_catalog')
9
10 class Catalog(models.Model):
11     id_catalog=models.AutoField('id_catalog',primary_key=True)
12     name=models.CharField(max_length=20,verbose_name="Каталог")
13
```

2)Делаем миграции (файл с миграциями создаётся автоматически):

```

models.py × 0001_initial.py × settings.py × serializers.py × views.py × urls.py ×
1 # Generated by Django 4.0 on 2021-12-19 13:31
2 from django.db import migrations, models
3 import django.db.models.deletion
4
5 class Migration(migrations.Migration):
6
7     initial = True
8
9     dependencies = [
10     ]
11
12     operations = [
13         migrations.CreateModel(
14             name='Catalog',
15             fields=[
16                 ('id_catalog', models.AutoField(primary_key=True, serialize=False, verbose_name='id_catalog')),
17                 ('namecat', models.CharField(max_length=20, verbose_name='Каталог')),
18             ],
19             options={
20             },
21         ),
22         migrations.CreateModel(
23             name='File',
24             fields=[
25                 ('id_file', models.AutoField(primary_key=True, serialize=False, verbose_name='id_file')),
26                 ('name', models.CharField(max_length=40, verbose_name='Название')),
27                 ('size', models.IntegerField(verbose_name='Размер')),
28                 ('extension', models.CharField(max_length=20, verbose_name='Расширение')),
29                 ('id_catalog', models.ForeignKey(db_column='id_catalog', on_delete=django.db.models.deletion.DO_NOTHING,
30
31 ]

```

3) Для возможности добавлять, удалять и изменять данные в таблицах был использован Django REST Framework:

Установим: `pip install djangorestframework` и добавим название модуля DRF в `INSTALLED_APPS` в файл `settings.py`:

```

models.py × 0001_initial.py × settings.py × serializers.py × views.py × urls.py ×
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'PK2.apps.Pk2Config',
41     'rest_framework'
42 ]

```

Написание сериализатора:

```
models.py × 0001_initial.py × settings.py × serializers.py × views.py × urls.py ×
1 from PK2.models import File, Catalog
2 from rest_framework import serializers
3
4 class FileSerializer(serializers.ModelSerializer):
5     class Meta:
6         # Модель, которую мы сериализуем
7         model = File
8         # Поля, которые мы сериализуем
9         fields = ["id_file", "name", "size", "extension", "id_catalog"]
10 class CatalogSerializer(serializers.ModelSerializer):
11     class Meta:
12         model = Catalog
13         fields = ["id_catalog", "name"]
```

Написание View:

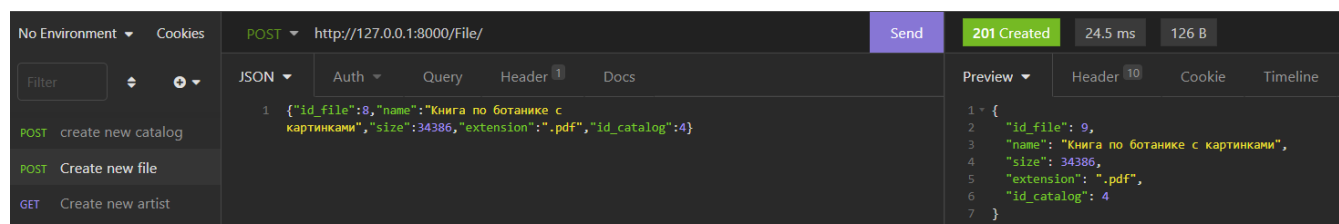
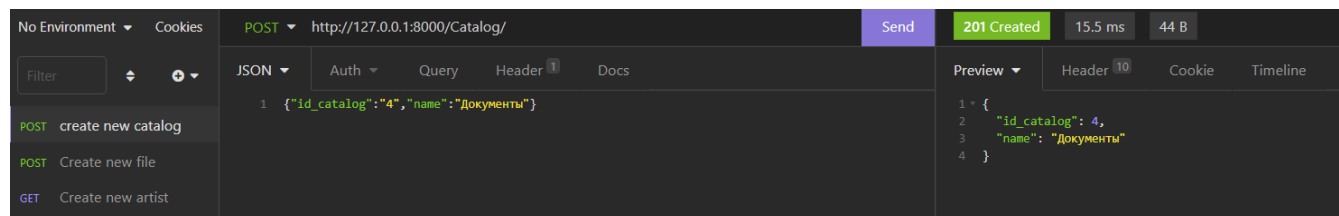
```
models.py × 0001_initial.py × settings.py × serializers.py × views.py × urls.py ×
1 from rest_framework import viewsets
2 from PK2.serializers import FileSerializer, CatalogSerializer
3 from PK2.models import File, Catalog
4
5 class FileViewSet(viewsets.ModelViewSet):
6     queryset = File.objects.all()
7     serializer_class = FileSerializer
8
9 class CatalogViewSet(viewsets.ModelViewSet):
10     queryset = Catalog.objects.all()
11     serializer_class = CatalogSerializer
12     # Create your views here.
13
```

Добавление View в URL'ы нашего приложения:

```
models.py × 0001_initial.py × settings.py × serializers.py × views.py × urls.py ×
13      1. Import the include() function: from django.urls import include, path
14      2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15      """
16      from django.contrib import admin
17      from PK2.views import FileViewSet, CatalogViewSet
18      from django.urls import include, path
19      from rest_framework import routers
20
21      router = routers.DefaultRouter()
22      router.register("File", FileViewSet)
23      router.register("Catalog", CatalogViewSet)
24
25      # Wire up our API using automatic URL routing.
26      # Additionally, we include login URLs for the browsable API.
27      urlpatterns = [
28          path('', include(router.urls)),
29          path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),
30          path('admin/', admin.site.urls),
31      ]
```

4) Для добавления, удаления и изменения данных в наших таблицах будем использовать приложение Insomnia:

Добавим данные в таблицы с помощью метода POST:



Просмотрим данные в наших таблицах с помощью метода GET:
Таблица Catalog:

Send	200 OK	17.8 ms	171 B	
	Preview ▾	Header 10	Cookie	Timeline
	<pre> 1 ▾ [2 ▾ { 3 "id_catalog": 1, 4 "name": "Изображения" 5 }, 6 ▾ { 7 "id_catalog": 2, 8 "name": "Музыка" 9 }, 10 ▾ { 11 "id_catalog": 3, 12 "name": "Видео" 13 }, 14 ▾ { 15 "id_catalog": 4, 16 "name": "Документы" 17 } 18]</pre>			

Таблица File:

Send	200 OK	10.2 ms	907 B	
	Preview ▾	Header 10	Cookie	Timeline
	<pre>1 ▾ [2 ▾ { 3 "id_file": 1, 4 "name": "Фотография бобра", 5 "size": 78, 6 "extension": ".png", 7 "id_catalog": 1 8 }, 9 ▾ { 10 "id_file": 2, 11 "name": "Фотографии с концерта", 12 "size": 367, 13 "extension": ".png", 14 "id_catalog": 1 15 }, 16 ▾ { 17 "id_file": 3, 18 "name": "Песни для танцев", 19 "size": 678, 20 "extension": ".mp3", 21 "id_catalog": 2 22 }, 23 ▾ { 24 "id_file": 4, 25 "name": "Фильмы романтические", 26 "size": 24568, 27 "extension": ".mp4", 28 "id_catalog": 3 29 }, 30 ▾ { 31 "id_file": 5, 32 "name": "Фильмы для семейного просмотра", 33 "size": 22543, 34 "extension": ".mp4", 35 "id_catalog": 3 36 },</pre>			

```

36 },
37 {
38   "id_file": 6,
39   "name": "Тексты песен Adele",
40   "size": 3456,
41   "extension": ".pdf",
42   "id_catalog": 4
43 },
44 {
45   "id_file": 7,
46   "name": "Текст презентации по итальянскому",
47   "size": 33696,
48   "extension": ".docx",
49   "id_catalog": 4
50 },
51 {
52   "id_file": 9,
53   "name": "Книга по ботанике с картинками",
54   "size": 34386,
55   "extension": ".pdf",
56   "id_catalog": 4
57 }
58 ]

```

Последняя запись имеет id_catalog=9, так как я удалила запись с id_catalog=8.

Изменим название файла и его размер для файла с id_file=7 с помощью метода PUT:

PUT http://127.0.0.1:8000/File/7/		Send	200 OK	482 ms	135 B
JSON	Auth	Query	Header 1	Docs	Preview
<pre>1 { "id_file": 7, "name": "Текст презентации по французскому", "size": 26886, "extension": ".pdf", "id_catalog": 4 }</pre>				<pre>1 { 2 "id_file": 7, 3 "name": "Текст презентации по французскому", 4 "size": 26886, 5 "extension": ".pdf", 6 "id_catalog": 4 7 }</pre>	

Удалим файл с id_file=9 с помощью метода DELETE:

DELETE http://127.0.0.1:8000/File/9/		Send	204 No Content	80.8 ms	0 B
JSON	Auth	Query	Header 1	Docs	Preview
<pre>1 ...</pre>				No body returned for response	

Проверим, что запись удалена:

```
200 OK 16.3 ms 781 B
Preview ▾ Header 10 Cookie Timeline
12     "size": 367,
13     "extension": ".png",
14     "id_catalog": 1
15 },
16 {
17     "id_file": 3,
18     "name": "Песни для танцев",
19     "size": 678,
20     "extension": ".mp3",
21     "id_catalog": 2
22 },
23 {
24     "id_file": 4,
25     "name": "Фильмы романтические",
26     "size": 24568,
27     "extension": ".mp4",
28     "id_catalog": 3
29 },
30 {
31     "id_file": 5,
32     "name": "Фильмы для семейного просмотра",
33     "size": 22543,
34     "extension": ".mp4",
35     "id_catalog": 3
36 },
37 {
38     "id_file": 6,
39     "name": "Тексты песен Adele",
40     "size": 3456,
41     "extension": ".pdf",
42     "id_catalog": 4
43 },
44 {
45     "id_file": 7,
46     "name": "Текст презентации по французскому",
47     "size": 26886,
48     "extension": ".pdf",
49     "id_catalog": 4
50 }
51 ]
```

5)Создадим представление и шаблон, формирующий отчёт, в котором будут содержаться данные из двух таблиц:

Представление GetFile, для вывода информации о связанной таблицы был использован `select_related` (связь один ко многим):


```
models.py × 0001_initial.py × settings.py × serializers.py × 0002_rename_namecat_catalog_name.py × base.html × datafromtables.html × views.py ×
1 from rest_framework import viewsets
2 from PK2.serializers import FileSerializer, CatalogSerializer
3 from PK2.models import File, Catalog
4 from django.shortcuts import render
5
6 class FileViewSet(viewsets.ModelViewSet):
7     queryset = File.objects.all()
8     serializer_class = FileSerializer
9
10 class CatalogViewSet(viewsets.ModelViewSet):
11     queryset = Catalog.objects.all()
12     serializer_class = CatalogSerializer
13
14 def GetFile(request):
15     context={
16         'data':File.objects.select_related('id_catalog')
17     }
18     return render(request, 'datafromtables.html', context)
```

Шаблоны:

Base.html

```
models.py × 0001_initial.py × settings.py × serializers.py × 0002_rename_namecat_catalog_name.py × base.html ×
1 <!doctype html>
2 <html lang="en" class="h-100">
3 <head>
4     <meta charset="utf-8">
5     <title>{% block title %}{% endblock %}</title>
6 </head>
7 <body>
8     {% block content %}{% endblock %}
9 </body>
10 </html>
```

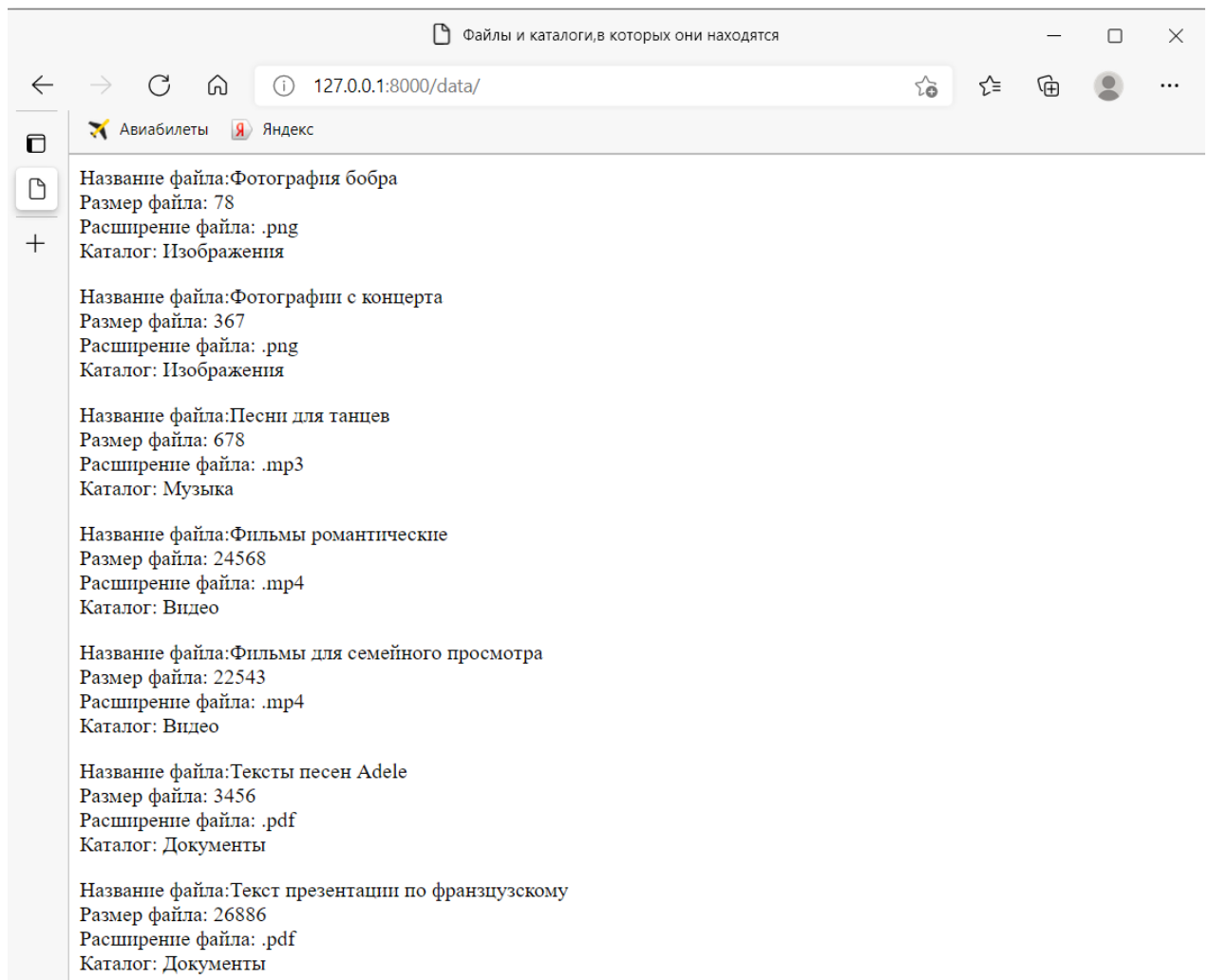
Datafromtables.html

```
models.py × 0001_initial.py × settings.py × serializers.py × 0002_rename_namecat_catalog_name.py × base.html × datafromtables.html ×
1 {% extends 'base.html' %}
2
3 {% block title %} Файлы и каталоги, в которых они находятся {% endblock %}
4
5 {% block content %}
6     {% for f in data %}
7         <div>Название файла: {{ f.name }}</div>
8         <div>Размер файла: {{ f.size }}</div>
9         <div>Расширение файла: {{ f.extension }}</div>
10        <div>Каталог: {{ f.id_catalog.name }}</div>
11        <p></p>
12    {% endfor %}
13 {% endblock %}
```

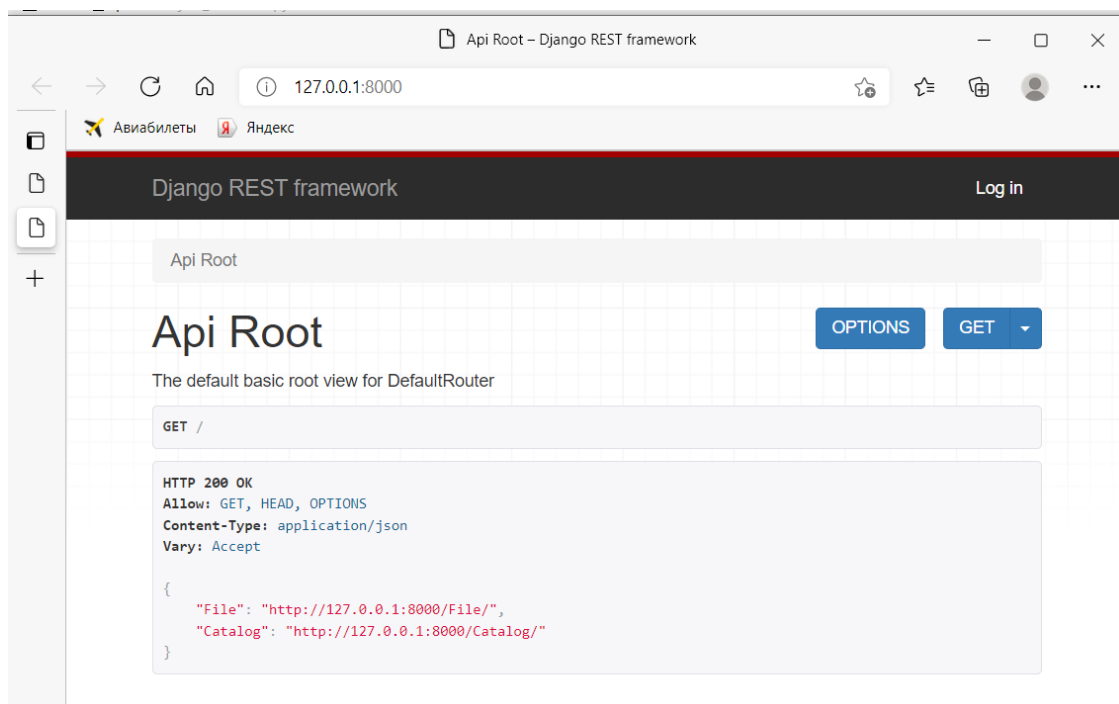
Добавим в `urls.py` путь для просмотра отчёта:

```
models.py × 0001_initial.py × settings.py × serializers.py × 0002_rename_namecat_catalog_name.py × base.html × datafromtables.html × views.py × urls.py ×
14 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from PK2.views import FileViewSet, CatalogViewSet, GetFile
18 from django.urls import include, path
19 from rest_framework import routers
20
21 router = routers.DefaultRouter()
22 router.register("File", FileViewSet)
23 router.register("Catalog", CatalogViewSet)
24
25
26 # Wire up our API using automatic URL routing.
27 # Additionally, we include login URLs for the browsable API.
28 urlpatterns = [
29     path('', include(router.urls)),
30     path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),
31     path('admin/', admin.site.urls),
32     path('data/', GetFile)
33 ]
```

В результате получаем такой отчёт:



Работа программы в отладочном интерфейсе DRF:



File List – Django REST framework

→ ↺ 🏠 ⓘ 127.0.0.1:8000/File/ ☆ ⚙ 📄 👤 ...

🛩️ Авиабилеты 🇷🇺 Яндекс

Django REST framework Log in

Api Root / File List

File List OPTIONS GET

GET /File/

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

```
[
  {
    "id_file": 1,
    "name": "Фотография бобра",
    "size": 78,
    "extension": ".png",
    "id_catalog": 1
  },
  {
    "id_file": 2,
    "name": "Фотографии с концерта",
    "size": 367,
    "extension": ".png",
    "id_catalog": 1
  },
  {
    "id_file": 3,
    "name": "Песни для танцев",
    "size": 678,
    "extension": ".mp3",
    "id_catalog": 2
  }
]
```

File List – Django REST framework

127.0.0.1:8000/File/

Авиабилеты Яндекс

Django REST framework Log in

```
},
{
  "id_file": 3,
  "name": "Песни для танцев",
  "size": 678,
  "extension": ".mp3",
  "id_catalog": 2
},
{
  "id_file": 4,
  "name": "Фильмы романтические",
  "size": 24568,
  "extension": ".mp4",
  "id_catalog": 3
},
{
  "id_file": 5,
  "name": "Фильмы для семейного просмотра",
  "size": 22543,
  "extension": ".mp4",
  "id_catalog": 3
},
{
  "id_file": 6,
  "name": "Тексты песен Adele",
  "size": 3456,
  "extension": ".pdf",
  "id_catalog": 4
},
{
  "id_file": 7,
  "name": "Текст презентации по французскому",
  "size": 26886,
  "extension": ".pdf",
  "id_catalog": 4
}
```

File List – Django REST framework

→ ↺ 🏠 ⓘ 127.0.0.1:8000/File/

☆

☆

🔒

👤

⋮

Авиабилеты Яandex

Django REST framework

Log in

```
[{"size": 3456, "extension": ".pdf", "id_catalog": 4}, {"id_file": 7, "name": "Текст презентации по французскому", "size": 26886, "extension": ".pdf", "id_catalog": 4}]
```

Raw data

HTML form

Название

Размер

Расширение

Id catalog

Catalog object (1) ▾

POST

Catalog List – Django REST framework

→ ↻ 🏠 ⓘ 127.0.0.1:8000/Catalog/ ⭐ ⚙️ 🔍 👤 ⋮

🛩️ Авиабилеты 🇷🇺 Яндекс

Django REST framework Log in

Api Root / Catalog List

Catalog List

OPTIONS GET ▾

GET /Catalog/

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

```
[
  {
    "id_catalog": 1,
    "name": "Изображения"
  },
  {
    "id_catalog": 2,
    "name": "Музыка"
  },
  {
    "id_catalog": 3,
    "name": "Видео"
  },
  {
    "id_catalog": 4,
    "name": "Документы"
  }
]
```

```
},
{
  "id_catalog": 4,
  "name": "Документы"
}
]
```

Raw data

HTML form

Каталог

POST