

**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:
студент группы ИУ5-52Б
Барышников Михаил

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Описание задания:

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

Создание сериализатора для модели ProgLanguage:

```
from computercomponents.models import Cpu
from rest_framework import serializers

class CompSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = Cpu
        # Поля, которые мы сериализуем
        fields = ["id", "name", "frequency", "architecture"]
```

Создание view:

```
class ComputeCompViewSet(viewsets.ModelViewSet):
    models = apps.get_app_config('computercomponents').get_models()
    model = ''
    for elem in models:
        if elem._meta.db_table == 'CPU':
            model = elem
    queryset = model.objects.all()
    serializer_class = CompSerializer
```

Добавление созданного view url's приложения:

```
from computercomponents import views as comp_views
from django.urls import include, path
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'computercomp', comp_views.ComputeCompViewSet)

urlpatterns = [
    path('api/', include(router.urls)),
    path('', comp_views.index),
    path('<str:model_name>/', comp_views.list, name='list')
]
```

Проверка работы API:

Запрос GET для получения всех элементов:

GET http://localhost:8000/api/computercomp/ Send 200 OK 704 ms 449 B

JSON Auth Query Header 1 Docs

1 | ...

Preview Header 7 Cookie Timeline

```
1 [
2   {
3     "id": 1,
4     "name": "Intel core i5",
5     "frequency": "5Ghz",
6     "architecture": "x86"
7   },
8   {
9     "id": 2,
10    "name": "Intel core i3",
11    "frequency": "3.2Ghz",
12    "architecture": "x86"
13  },
14  {
15    "id": 3,
16    "name": "Apple M1 PRO",
17    "frequency": "3.9Mhz",
18    "architecture": "arm"
19  },
20  {
21    "id": 4,
22    "name": "Ryzen 5",
23    "frequency": "4.6Ghz",
24    "architecture": "x86"
25  }
26 ]
```

Запрос GET для получения 1 элемента:

GET http://localhost:8000/api/computercomp/3 Send 200 OK 32.4 ms 111 B

JSON Auth Query Header 1 Docs

1 | ...

Preview Header 7 Cookie Timeline

```
1 {
2   "id": 3,
3   "name": "Apple M1 PRO",
4   "frequency": "3.9Mhz",
5   "architecture": "arm"
6 }
```

Запрос POST добавления нового элемента:

POST http://localhost:8000/api/computercomp/ Send 201 Created 26.2 ms 65 B

JSON Auth Query Header 1 Docs

1 {
2 "name": "Ryzen 4",
3 "frequency": "3Mhz",
4 "architecture": "x86"
5 }

Preview Header 7 Cookie

```
1 {
2   "id": 5,
3   "name": "Ryzen 4",
4   "frequency": "3Mhz",
5   "architecture": "x86"
6 }
```

Запрос DELETE для удаления элемента:

The screenshot shows a REST client interface. The top bar displays the method **DELETE** and the URL `http://localhost:8000/api/computercomp/5/`. A **Send** button is on the right. The response status is **204 No Content** in a green box, with a response time of **23.1 ms** and a size of **0 B**. Below the top bar, there are tabs for **JSON**, **Auth**, **Query**, **Header** (with a count of 1), and **Docs**. The **JSON** tab is active, showing the request body as a JSON object:

```
1 {
2   "name": "Ryzen 4",
3   "frequency": "3Mhz",
4   "architecture": "x86"
5 }
```

 On the right side, there are tabs for **Preview**, **Header** (with a count of 6), and **Cookie**. The **Preview** tab is active, displaying the text: **No body returned for response**.

Запрос GET для получения всех элементов:

The screenshot shows a REST client interface. The top bar displays the method **GET** and the URL `http://localhost:8000/api/computercomp`. A **Send** button is on the right. The response status is **200 OK** in a green box, with a response time of **22.1 ms** and a size of **449 B**. Below the top bar, there are tabs for **JSON**, **Auth**, **Query**, **Header** (with a count of 1), and **Docs**. The **JSON** tab is active, showing the request body as a JSON object:

```
1 {
2   "name": "Ryzen 4",
3   "frequency": "3Mhz",
4   "architecture": "x86"
5 }
```

 On the right side, there are tabs for **Preview**, **Header** (with a count of 7), **Cookie**, and **Timeline**. The **Preview** tab is active, displaying a JSON array of four computer components:

```
1 [
2   {
3     "id": 1,
4     "name": "Intel core i5",
5     "frequency": "5Ghz",
6     "architecture": "x86"
7   },
8   {
9     "id": 2,
10    "name": "Intel core i3",
11    "frequency": "3.2Ghz",
12    "architecture": "x86"
13  },
14  {
15    "id": 3,
16    "name": "Apple M1 PRO",
17    "frequency": "3.9Mhz",
18    "architecture": "arm"
19  },
20  {
21    "id": 4,
22    "name": "Ryzen 5",
23    "frequency": "4.6Ghz",
24    "architecture": "x86"
25  }
26 ]
```