



**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 «Радиотехнический»

Кафедра ИУ5 «Системы обработки информации и управления»

Лабораторная работа №6

по дисциплине «Разработка интернет-приложений»

**Выполнил:
студент группы РТ5-51Б
М.А. Ходосов**

2021 г.

Задание лабораторной работы:

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

Текст программы:

Исходный код программы доступен по ссылке в github-репозитории - [link](#).

Экранные формы с примерами выполнения программы:

Django REST framework

Log in

Api Root

OPTIONSGET

Api Root

The default basic root view for DefaultRouter

GET /

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

```
{
  "Operator": "http://127.0.0.1:8000/Operator/",
  "Proglang": "http://127.0.0.1:8000/Proglang/"
}
```

Oper List

OPTIONSGET

GET /Operator/

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

```
[
  {
    "id": 1,
    "name": "!",
    "count_cycles": 100
  },
  {
    "id": 2,
    "name": "&&",
    "count_cycles": 200
  },
  {
    "id": 5,
    "name": "- ",
    "count_cycles": 170
  },
  {
    "id": 4,
    "name": "<<",
    "count_cycles": 50
  },
  {
    "id": 3,
    "name": "||",
    "count_cycles": 200
  }
]
```

Raw data

HTML form

Название оператора

Raw dataHTML form

Название оператора

Количество тактов ЦП

POST

GET /Operator/

HTTP 200 OK

Allow: GET, POST, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
[
  {
    "id": 1,
    "name": "!",
    "count_cycles": 100
  },
  {
    "id": 2,
    "name": "&&",
    "count_cycles": 200
  },
  {
    "id": 5,
    "name": "-.",
    "count_cycles": 170
  },
  {
    "id": 4,
    "name": "<<",
    "count_cycles": 50
  },
  {
    "id": 6,
    "name": "A",
    "count_cycles": 90
  },
  {
    "id": 3,
    "name": "||",
    "count_cycles": 200
  }
]
```

Oper Instance

DELETEOPTIONSGET

GET /Operator/2/

HTTP 200 OK

Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
{
  "id": 2,
  "name": "&&",
  "count_cycles": 200
}
```

Raw dataHTML form

Название оператора

Количество тактов ЦП

PUT

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

```
[
  {
    "id": 1,
    "name": "!",
    "count_cycles": 100
  },
  {
    "id": 2,
    "name": "&&",
    "count_cycles": 200
  }
]
```

Raw data

HTML form

Media type: application/json

Content:

```
{
  "title": "Swift - для любителей яблок"
}
```

POST

POST /Proglang/

HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

```
{
  "id": 4,
  "title": "Swift - для любителей яблок"
}
```