

**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  
Васильченко Дарья

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 IT_company.models import ITCompany
from rest_framework import serializers

class ITCompanySerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = ITCompany
        # Поля, которые мы сериализуем
        fields = ["id_company", "name_company", "foundation_year",
                  "specialization"]
```

## Создание view:

```
from rest_framework import viewsets
from IT_company.serializers import ITCompanySerializer
from IT_company.models import ITCompany

class ITCompanyViewSet(viewsets.ModelViewSet):
    """
    API endpoint, который позволяет просматривать и редактировать информацию об
    IT-компаниях
    """
    # queryset всех пользователей для фильтрации по дате последнего изменения
    queryset = ITCompany.objects.all().order_by('name_company')
    serializer_class = ITCompanySerializer # Сериализатор для модели
```

## Добавление созданного view url's приложения:

```
from django.contrib import admin
from IT_company import views as it_company_views
from django.urls import include, path
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'it_company', it_company_views.ITCompanyViewSet)

# Wire up our API using automatic URL routing.
# Additionally, we include login URLs for the browsable API.
urlpatterns = [
    path('', include(router.urls)),
    path('api-auth/', include('rest_framework.urls',
```

```
namespace='rest_framework')),\n\n    path('admin/', admin.site.urls),\n]\n
```

## Проверка работы API:

Запрос GET для получения всех элементов:

GET http://127.0.0.1:8000/it\_company/ Send 200 OK 5.36 ms 211 B Just Now

JSON Auth Query Header 1 Docs

```
1 ...
```

Preview Header 10 Cookie Timeline

```
1 - [\n2 - {\n3 -   "id_company": 1,\n4 -   "name_company": "Dell Technologiessss",\n5 -   "foundation_year": 1984,\n6 -   "specialization": "Production of computers"\n7 - },\n8 - {\n9 -   "id_company": 5,\n10 -  "name_company": "IBM",\n11 -  "foundation_year": 1911,\n12 -  "specialization": "Software"\n13 - }\n14 - ]
```

Запрос GET для получения 1 элемента:

GET http://127.0.0.1:8000/it\_company/1 Send 200 OK 20.5 ms 120 B Just Now

JSON Auth Query Header 1 Docs

```
1 ...
```

Preview Header 10 Cookie Timeline

```
1 - {\n2 -   "id_company": 1,\n3 -   "name_company": "Dell Technologiessss",\n4 -   "foundation_year": 1984,\n5 -   "specialization": "Production of computers"\n6 - }
```

Запрос PUT для изменения элемента:

PUT http://127.0.0.1:8000/it\_company/1/ Send 200 OK 30.2 ms 117 B Just Now

JSON Auth Query Header 1 Docs

```
1 {"name_company": "Dell Technologies"}
```

Preview Header 10 Cookie Timeline

```
1 - {\n2 -   "id_company": 1,\n3 -   "name_company": "Dell Technologies",\n4 -   "foundation_year": 1984,\n5 -   "specialization": "Production of computers"\n6 - }
```

Запрос POST добавления нового элемента:

POST http://127.0.0.1:8000/it\_company/ Send 201 Created 20 ms 88 B Just Now

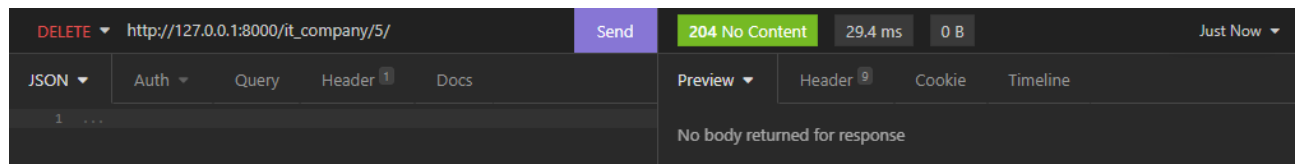
JSON Auth Query Header 1 Docs

```
1 - {\n2 -   "name_company": "IBM",\n3 -   "foundation_year": 1911,\n4 -   "specialization": "Software"\n5 - }\n
```

Preview Header 10 Cookie Timeline

```
1 - {\n2 -   "id_company": 6,\n3 -   "name_company": "IBM",\n4 -   "foundation_year": 1911,\n5 -   "specialization": "Software"\n6 - }
```

Запрос DELETE для удаления элемента:



Запрос GET для получения всех элементов:

