

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

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

Курс «Разработка интернет-приложений»
Отчет по лабораторной работе №6

Выполнил:
студент группы РТ5-51Б

Грызин Алексей

Подпись и дата:

Проверил:
преподаватель каф.
ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Описание задания

Цель лабораторной работы: изучение возможностей разработки REST API с использованием Django REST Framework.

Задание:

В этой лабораторной работе Вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также Вам нужно будет дополнить свои классы предметной области, связав их с созданной БД. После этого Вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей.

1. Создайте сценарий с подключением к БД и несколькими запросами, примеры рассмотрены в методических указаниях.
2. Реализуйте модели Вашей предметной области из предыдущей ЛР (минимум две модели, т.е. две таблицы).
3. Создайте представления и шаблоны Django для отображения списка данных по каждой из сущностей.

Ход работы

```
settings.py U  __init__.py 2  TC New Request  serializers.py U x  views.py U
lab6 > django_orm > master > serializers.py > ProcSerializer
1  from rest_framework import serializers
2  from .models import Processor
3
4  class ProcSerializer(serializers.ModelSerializer):
5      class Meta:
6          # Модель, которую мы сериализуем
7          model = Processor
8          # Поля, которые мы сериализуем
9          fields = ["id", "title", "numb_cores", "freq", "price", "year"]
```

```
class ProcViewSet(viewsets.ModelViewSet):
    '''API endpoint'''
    queryset = Processor.objects.all()
    serializer_class = ProcSerializer
```

Результат работы программы

GET

http://127.0.0.1:8000/processor

Send

Query

Headers²

Auth

Body

Tests

Query Parameters

☐

parameter

value

Status: 200 OKSize: 256 BytesTime: 11 ms

Response

Headers⁹

Cookies

Test Results

1

=

{

2

=

{

3

"id": 1,

4

"title": "AMD FX 4300",

5

"numb_cores": 4,

6

"freq": 3800,

7

"price": 4000,

8

"year": 2021

9

},

10

=

{

11

"id": 2,

12

"title": "AMD FX 8300",

13

"numb_cores": 8,

14

"freq": 3500,

15

"price": 8000,

16

"year": 2021

17

},

18

=

{

19

"id": 3,

20

"title": "AMD Ryzen 5 3600",

21

"numb_cores": 6,

22

"freq": 3700,

23

"price": 12000,

24

"year": 2021

25

}

26

}

GET

http://127.0.0.1:8000/processor/1

Send

Query

Headers²

Auth

Body

Tests

Query Parameters

☐

parameter

value

Status: 200 OKSize: 82 BytesTime: 8 ms

Response

Headers⁹

Cookies

Test Results

1

=

{

2

"id": 1,

3

"title": "AMD FX 4300",

4

"numb_cores": 4,

5

"freq": 3800,

6

"price": 4000,

7

"year": 2021

8

}

POST

http://127.0.0.1:8000/processor/

Send

Query

Headers²

Auth

Body¹

Tests

Json

Xml

Text

Form

Form-encode

GraphQL

Binary

Form Fields

☐

Files

☒

title

Intel Core i7-4790

☒

numb_cores

8

☒

freq

3600

☒

price

8000

☒

year

2014

☐

field name

value

Status: 201 CreatedSize: 89 BytesTime: 23 ms

Response

Headers⁹

Cookies

Test Results

1

=

{

2

"id": 4,

3

"title": "Intel Core i7-4790",

4

"numb_cores": 8,

5

"freq": 3600,

6

"price": 8000,

7

"year": 2014

8

}

GET

http://127.0.0.1:8000/processor/

Send

Query

Headers²

Auth

Body¹

Tests

Json

Xml

Text

Form

Form-encode

GraphQL

Binary

Form Fields

☐

Files

☒

title

Intel Core i7-4790

☒

numb_cores

8

☒

freq

3600

☒

price

8000

☒

year

2014

☐

field name

value

Status: 200 OKSize: 346 BytesTime: 9 ms

Response

Headers⁹

Cookies

Test Results

1

=

{

2

=

{

3

"id": 1,

4

"title": "AMD FX 4300",

5

"numb_cores": 4,

6

"freq": 3800,

7

"price": 4000,

8

"year": 2021

9

},

10

=

{

11

"id": 2,

12

"title": "AMD FX 8300",

13

"numb_cores": 8,

14

"freq": 3500,

15

"price": 8000,

16

"year": 2021

17

},

18

=

{

19

"id": 3,

20

"title": "AMD Ryzen 5 3600",

21

"numb_cores": 6,

22

"freq": 3700,

23

"price": 12000,

24

"year": 2021

25

}

26

=

{

27

"id": 4,

28

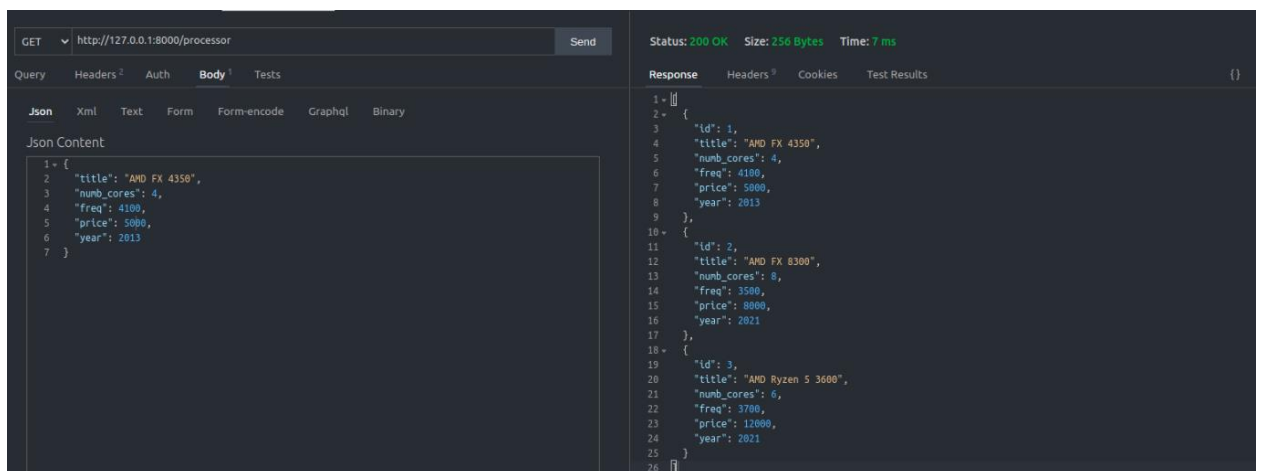
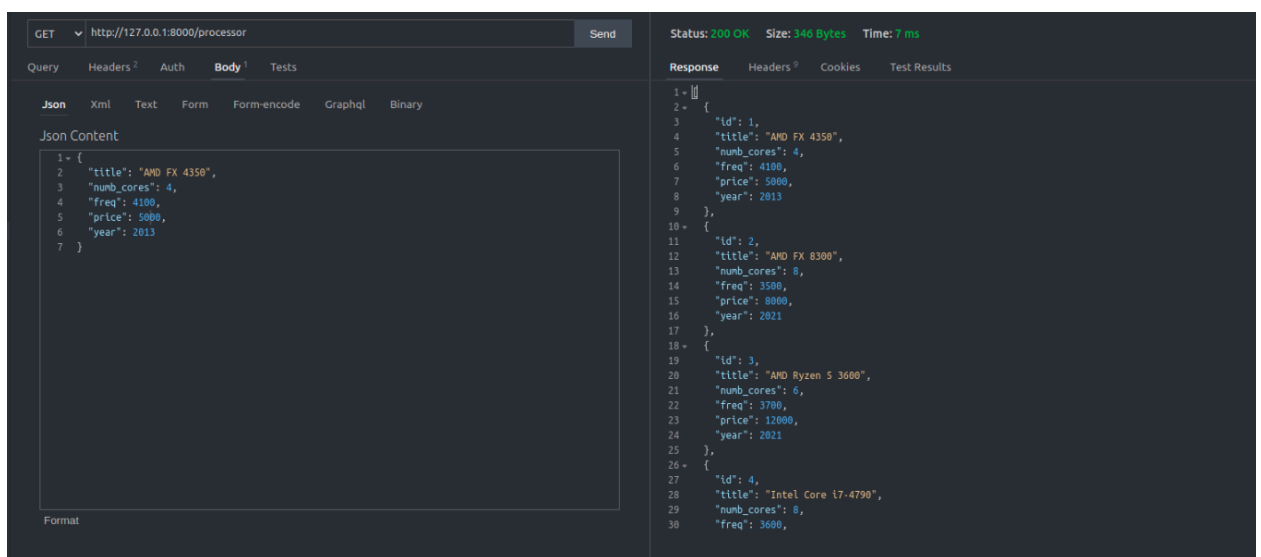
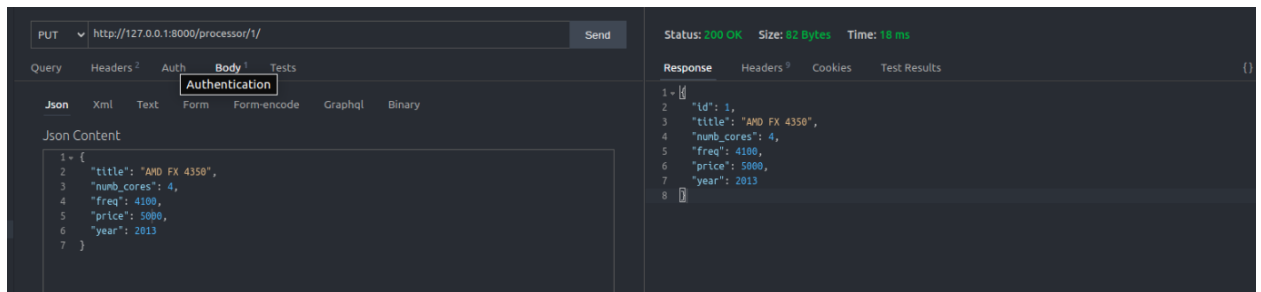
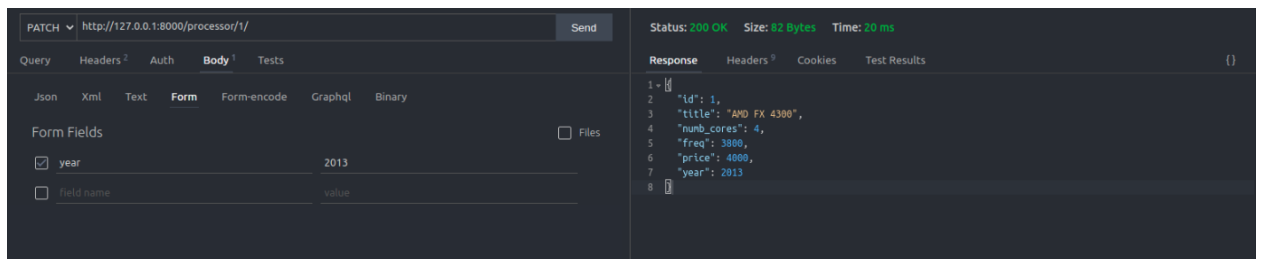
"title": "Intel Core i7-4790",

29

"numb_cores": 8,

30

"freq": 3600,



Текст программы

Django-проект находится в этой папке репозитория GitHub:

<https://github.com/feelsbadmans/bmstu-5-sem-rip/tree/main/lab6>