

**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-13**

Выполнил:
студент группы ИУ5-53Б
Кузнецов Г.И.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

Описание предметной области:

Существуют две сущности: Книга и Библиотека.

Книга имеет следующие атрибуты: номер (id), фамилия автора (author_name), количество страниц (page_count), жанр (genre) и номер связанной библиотеки для связи один-ко-многим (lib_id).

Библиотека имеет следующие атрибуты: номер (id) и название (name).

Библиотека и книга связаны отношением один-ко-многим. Была создана соответствующая база данных MySQL

Задание моделей Django ORM:

```
class Book(models.Model):
    idbook = models.AutoField(primary_key=True)
    author_name = models.CharField(max_length=30, blank=True, null=True)
    pages_count = models.IntegerField(blank=True, null=True)
    genre = models.CharField(max_length=15, blank=True, null=True)
    idlibrary = models.ForeignKey('Library', models.DO_NOTHING,
db_column='idlibrary', blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'book'

class Library(models.Model):
    idlibrary = models.AutoField(primary_key=True)
    name = models.CharField(max_length=30, blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'library'
```

Для возможности добавления, редактирования и удаления данных были использован rest_framework.

Сериализаторы:

```
class LibrarySerializer(serializers.ModelSerializer):
    class Meta:
        model = models.Library
        fields = ["idlibrary", "name"]

class BookSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.Book
        fields = ["idbook", "author_name", "pages_count", "genre",
"idlibrary"]
```

Представления:

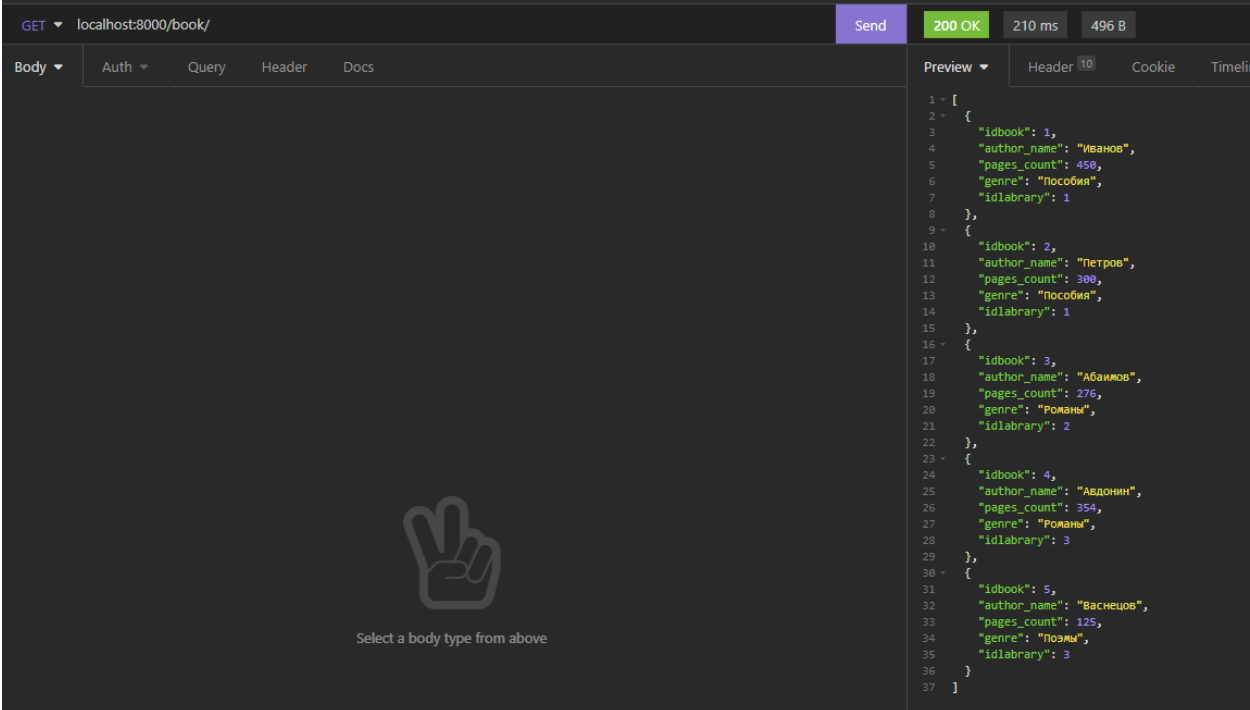
```
class LibraryViewSet(viewsets.ModelViewSet):
    queryset = models.Library.objects.all()
    serializer_class = serializers.LibrarySerializer

class BookViewSet(viewsets.ModelViewSet):
    queryset = models.Book.objects.all()
    serializer_class = serializers.BookSerializer
```

Route'инг:

```
router = routers.DefaultRouter()
router.register('library', views.LibraryViewSet)
router.register('book', views.BookViewSet)
```

Пример выполнения запросов:

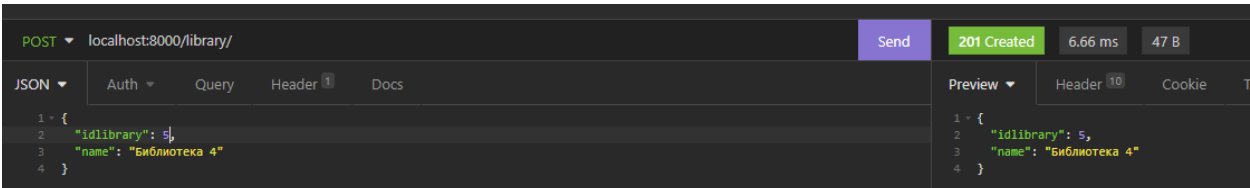


GET localhost:8000/book/ Send 200 OK 210 ms 496 B

Body Auth Query Header Docs

Preview Header 10 Cookie Timeli

```
1 - [
2 - {
3 -   "idbook": 1,
4 -   "author_name": "Иванов",
5 -   "pages_count": 450,
6 -   "genre": "Пособия",
7 -   "idlibrary": 1
8 - },
9 - {
10 -   "idbook": 2,
11 -   "author_name": "Петров",
12 -   "pages_count": 300,
13 -   "genre": "Пособия",
14 -   "idlibrary": 1
15 - },
16 - {
17 -   "idbook": 3,
18 -   "author_name": "Абакиров",
19 -   "pages_count": 270,
20 -   "genre": "Романы",
21 -   "idlibrary": 2
22 - },
23 - {
24 -   "idbook": 4,
25 -   "author_name": "Авдонин",
26 -   "pages_count": 354,
27 -   "genre": "Романы",
28 -   "idlibrary": 3
29 - },
30 - {
31 -   "idbook": 5,
32 -   "author_name": "Васнецов",
33 -   "pages_count": 125,
34 -   "genre": "Поэмы",
35 -   "idlibrary": 3
36 - }
37 ]
```



POST localhost:8000/library/ Send 201 Created 6.66 ms 47 B

JSON Auth Query Header 1 Docs

Preview Header 10 Cookie T

```
1 - {
2 -   "idlibrary": 5,
3 -   "name": "Библиотека 4"
4 - }
```

Для создания отчёта, соединяющего данные двух таблиц было создано следующее представление:

```
def getBooks(library: models.Library) -> list[models.Book]:
    result = []
    for book in models.Book.objects.all():
        if book.idlibrary == library:
            result.append(book)
    return result

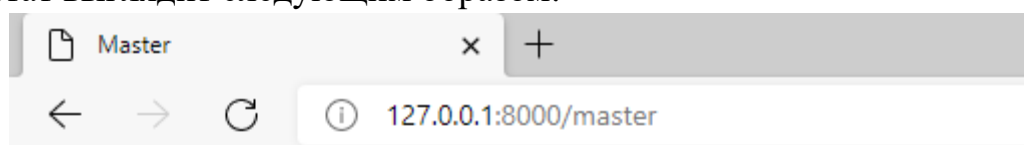
def master(request):
    libraries = models.Library.objects.all()
    values = []
    for library in libraries:
        values.append((library, getBooks(library)))

    return render(request, "master.html", {"values": values })
```

И шаблон:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Master</title>
  <style>
    table.collapse{
      border: 2px solid black;
      border-collapse: collapse;
      margin-top: 10px;
    }
    td.upper{
      border: 2px solid black;
      padding: 10px;
    }
    td{
      border: 2px solid black;
      padding: 5px;
    }
  </style>
</head>
<body>
{% for value in values %}
  <table class="collapse">
    <tr>
      <td class="upper" colspan="2">ID библиотеки</td>
      <td class="upper" colspan="2">Название библиотеки</td>
    </tr>
    <tr>
      <td class="upper" colspan="2">{{ value.0.idlibrary }}</td>
      <td class="upper" colspan="2">{{ value.0.name }}</td>
    </tr>
    <tr>
      <td>ID книги</td>
      <td>Автор книги</td>
      <td>Количество страниц</td>
      <td>Жанр</td>
    </tr>
    {% for book in value.1 %}
      <tr>
        <td>{{ book.idbook }}</td>
        <td>{{ book.author_name }}</td>
        <td>{{ book.pages_count }}</td>
        <td>{{ book.genre }}</td>
      </tr>
    {% endfor %}
  </table>
{% endfor %}
</body>
</html>
```

Результат выглядит следующим образом:



ID библиотеки		Название библиотеки	
1		Библиотека 1	
ID книги	Автор книги	Количество страниц	Жанр
1	Иванов	450	Пособия
2	Петров	300	Пособия

ID библиотеки		Название библиотеки	
2		Библиотека 2	
ID книги	Автор книги	Количество страниц	Жанр
3	Абаимов	276	Романы

ID библиотеки		Название библиотеки	
3		Библиотека 3	
ID книги	Автор книги	Количество страниц	Жанр
4	Авдонин	354	Романы
5	Васнецов	125	Поэмы