

Задание

Вариант	Предметная область	
	Класс 1	Класс 2
26	Раздел	Документ

1. Создайте проект Python Django с использованием стандартных средств Django.
2. Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

Исходный код

settings.py

```
"""
Django settings for RK_2 project.

Generated by 'django-admin startproject' using Django 4.0.

For more information on this file, see
https://docs.djangoproject.com/en/4.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.0/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-x06c#51*5efslu(ypwc93jb=ue32a)t@e_tnp4pg402i@9)raz'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
```

```

'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'rest_framework',
'App.apps.AppConfig',
]

MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'RK_2.urls'

TEMPLATES = [
{
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
    'DIRS': [BASE_DIR / 'templates']
    ,
    'APP_DIRS': True,
    'OPTIONS': {
        'context_processors': [
            'django.template.context_processors.debug',
            'django.template.context_processors.request',
            'django.contrib.auth.context_processors.auth',
            'django.contrib.messages.context_processors.messages',
        ],
    },
},
]

WSGI_APPLICATION = 'RK_2.wsgi.application'

# Database
# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {

```

```

        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/4.0/topics/i18n/

LANGUAGE_CODE = 'ru-ru'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

models.py

```

from django.db import models

# Create your models here.
class Course(models.Model):
    number = models.CharField('Номер документа', max_length=50)

class Group(models.Model):
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    number = models.CharField('Номер раздела', max_length=50)

```

serializers.py

```

from rest_framework import serializers
from .models import Course, Group

class CourseSerializer(serializers.ModelSerializer):
    class Meta:
        model = Course
        fields = ['id', 'number']

class GroupSerializer(serializers.ModelSerializer):
    class Meta:
        model = Group
        fields = ['id', 'course', 'number']

```

views.py

```

from rest_framework import viewsets
from .serializers import CourseSerializer, GroupSerializer

```

```

from .models import Course, Group
from django.shortcuts import render
from django.http import Http404

# Create your views here.
class CourseViewSet(viewsets.ModelViewSet):
    queryset = Course.objects.all()
    serializer_class = CourseSerializer

class GroupViewSet(viewsets.ModelViewSet):
    queryset = Group.objects.all()
    serializer_class = GroupSerializer

def CourseList(request):
    return render(request, 'course/list.html', {'c_list': Course.objects.all()})

def GroupList(request):
    return render(request, 'group/list.html', {'g_list': Group.objects.all()})

def CourseView(request, course_id=0):
    try:
        c = Course.objects.get(id=course_id)
    except:
        raise Http404('Документ не найден!')
    return render(request, 'course/detail.html', {'course': c})

def GroupView(request, group_id):
    try:
        g = Group.objects.get(id=group_id)
    except:
        raise Http404('Раздел не найден!')
    return render(request, 'group/detail.html', {'group': g})

```

urls.py

```

"""RK_2 URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include
from rest_framework import routers
from App import views as app_views

router = routers.DefaultRouter()
router.register(r'courses', app_views.CourseViewSet)
router.register(r'groups', app_views.GroupViewSet)

```

```
# Wire up our API using automatic URL routing.
# Additionally, we include login URLs for the browsable API.
urlpatterns = [
    path('', include(router.urls)),
    path('admin/', admin.site.urls),
    path('app/c_list', app_views.CourseList),
    path('app/g_list', app_views.GroupList),
    path('app/c/<int:course_id>', app_views.CourseView, name='course_detail'),
    path('app/g/<int:group_id>', app_views.GroupView, name='group_detail')
]
```

base.html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}Рубежный контроль № 2{% endblock %}</title>
</head>
<body>

{% block content %}{% endblock %}

<hr>
<center>Рубин Р. О. ИУ5Ц-72Б</center>

</body>
</html>
```

course/list.html

```
{% extends 'base.html' %}

{% block title %}PK №2{% endblock %}

{% block content %}

    <center><h1>Список документов</h1></center><hr><br>

    {% if c_list %}
        {% for c in c_list %}
            <a href="{% url 'course_detail' c.id %}">{{ c.number }}</a><br>
        {% endfor %}
    {% else %}
        Список пуст
    {% endif %}

{% endblock %}
```

course/detail.html

```
{% extends 'base.html' %}

{% block title %}{{ course.number }}{% endblock %}

{% block content %}

    <center><h1>{{ course.number }}</h1></center><hr><br>

    <table border="1" cellpadding="5">
    <tr>
        <th>Номер документа</th>
        <th>{{ course.number }}</th>
    </tr>
    </table>
    <br><hr><br>
    <a href="/app/c_list">Назад</a>
```

```
{% endblock %}
```

group/list.html

```
{% extends 'base.html' %}

{% block title %}PK №2{% endblock %}

{% block content %}

    <center><h1>Список разделов</h1></center><hr><br>

    {% if g_list %}
        {% for g in g_list %}
            <a href="{% url 'group_detail' g.id %}">{{ g.number }}</a><br>
        {% endfor %}
    {% else %}
        Список пуст
    {% endif %}

{% endblock %}
```

group/detail.html

```
{% extends 'base.html' %}

{% block title %}{{ group.number }}{% endblock %}

{% block content %}

    <center><h1>{{ group.number }}</h1></center><hr><br>

    <table border="1" cellpadding="5">
    <tr>
        <th>Документ</th>
        <th>{{ group.course.number }}</th>
    </tr>
    <tr>
        <th>Раздел</th>
        <th>{{ group.number }}</th>
    </tr>
    </table>
    <br><hr><br>
    <a href="/app/g_list">Назад</a>

{% endblock %}
```

Добавление, изменение и удаление данных через API

Добавление документа

Insomnia - LAB6 - create new stock

ApplicationEditViewWindowToolsHelp

Insomnia / LAB6

No EnvironmentCookies

POSThttp://127.0.0.1:8000/courses/Send

201 Created786 ms22 BJust Now

Filter

POSTcreate new stock

JSONAuthQueryHeader1Docs

```
1 * {
2   "number": 4
3 }
```

PreviewHeader10CookieTimeline

```
1 * {
2   "id": 11,
3   "number": "4"
4 }
```

Удаление документа

Insomnia - LAB6 - create new stock

ApplicationEditViewWindowToolsHelp

Insomnia / LAB6

No EnvironmentCookies

DELETEhttp://127.0.0.1:8000/courses/10Send

204 No Content537 ms0 BJust Now

Filter

DELcreate new stock

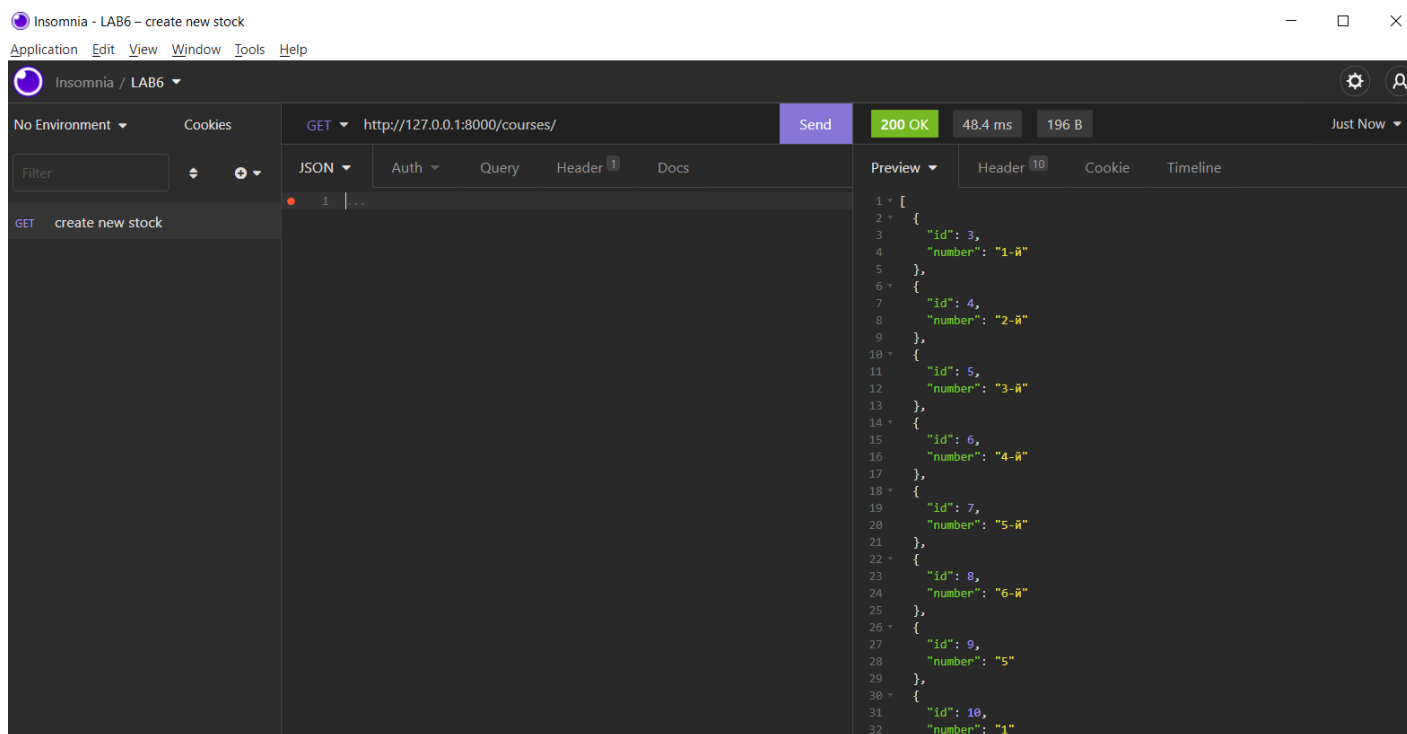
JSONAuthQueryHeader1Docs

```
1 ...
```

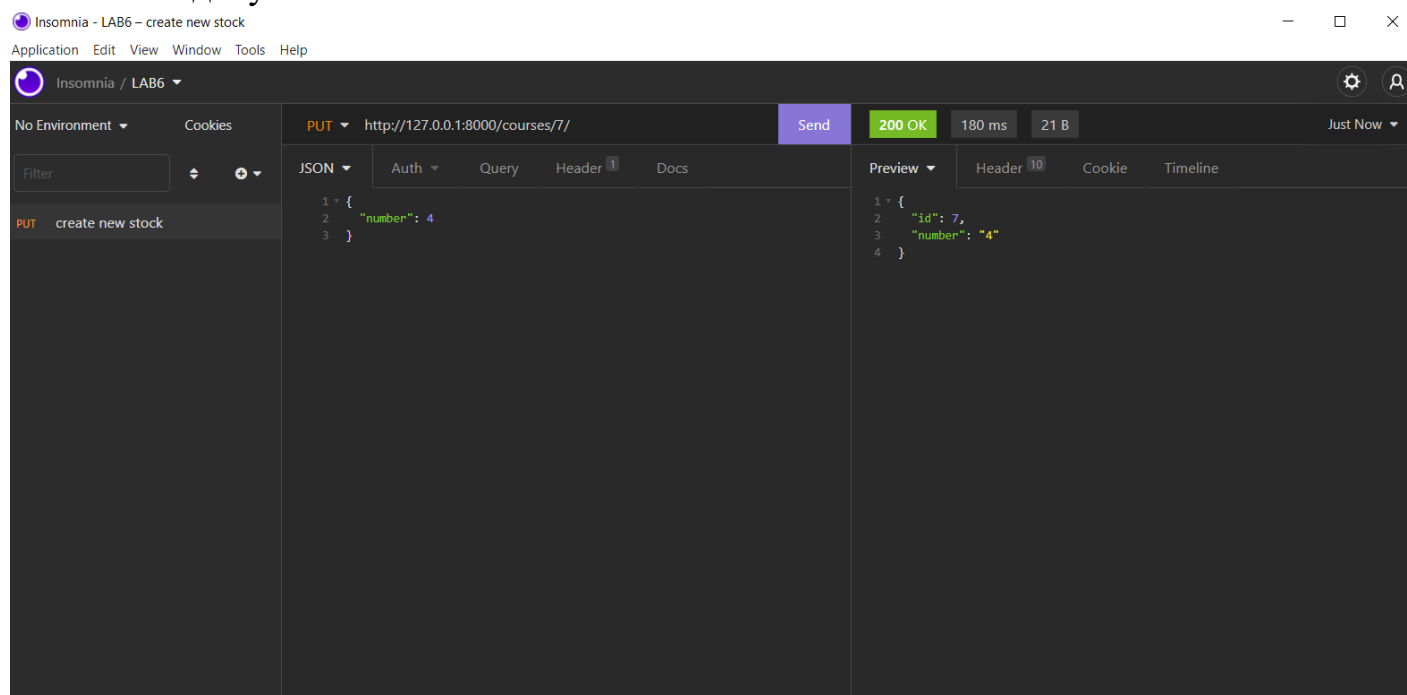
PreviewHeader9CookieTimeline

No body returned for response

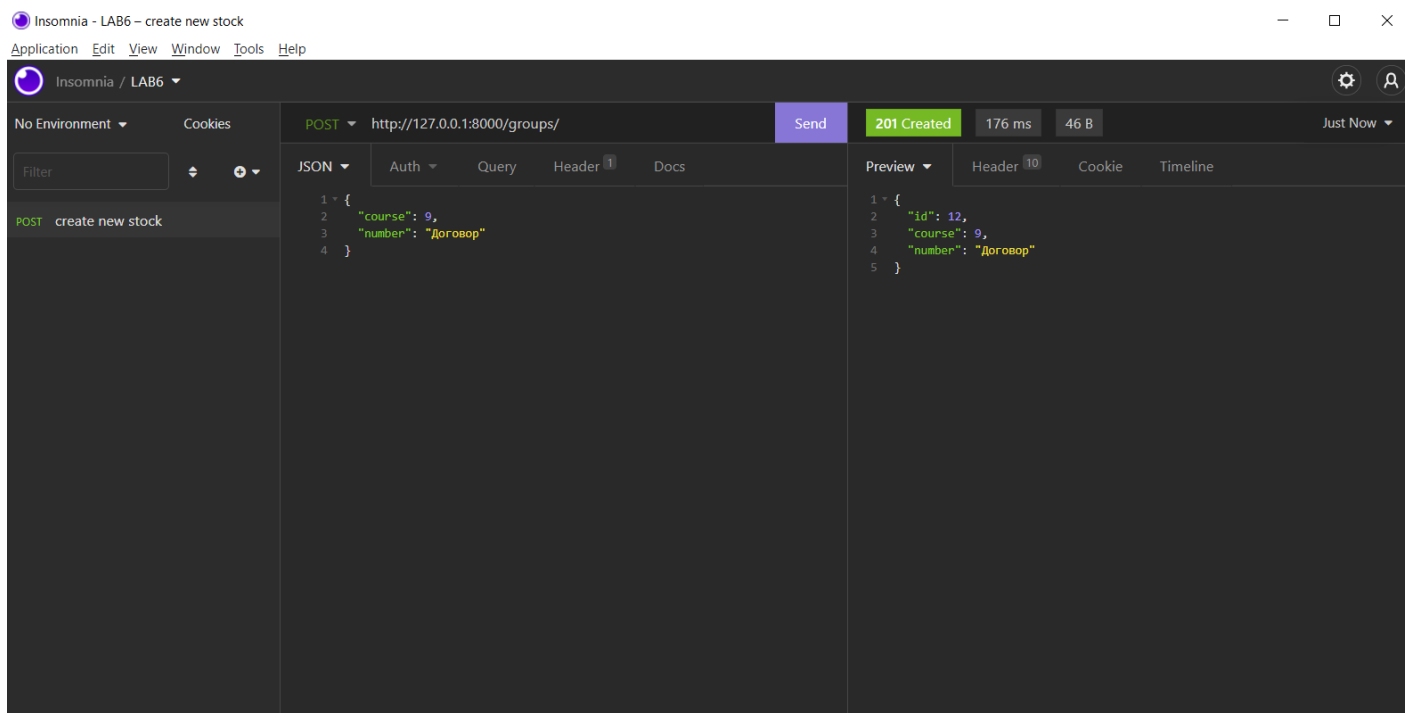
Список документов



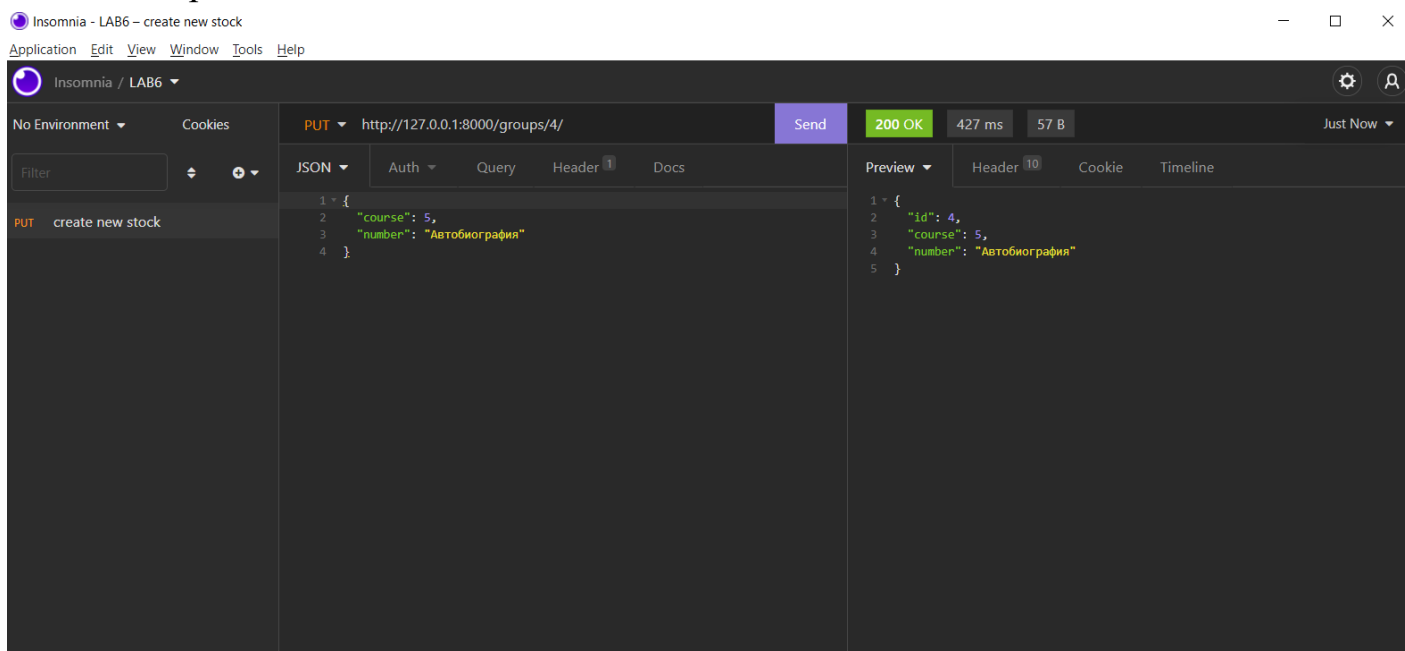
Изменение документа



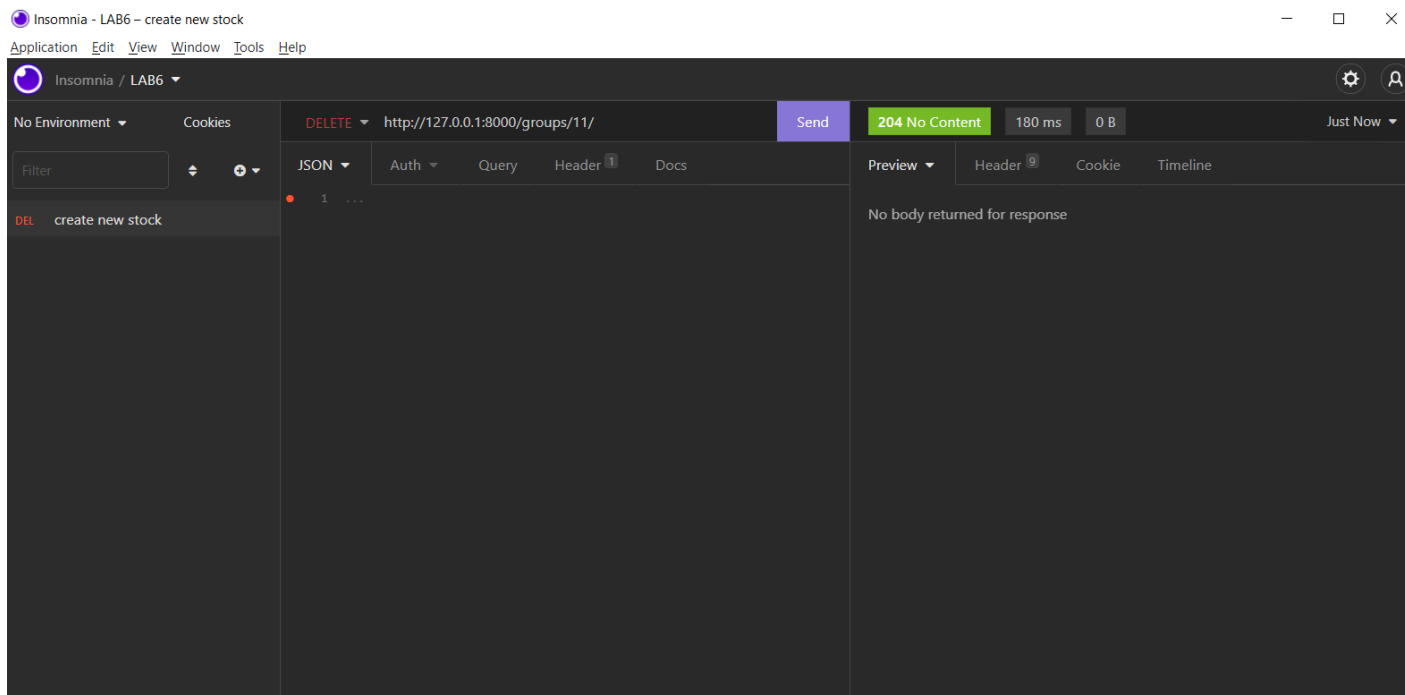
Добавление раздела



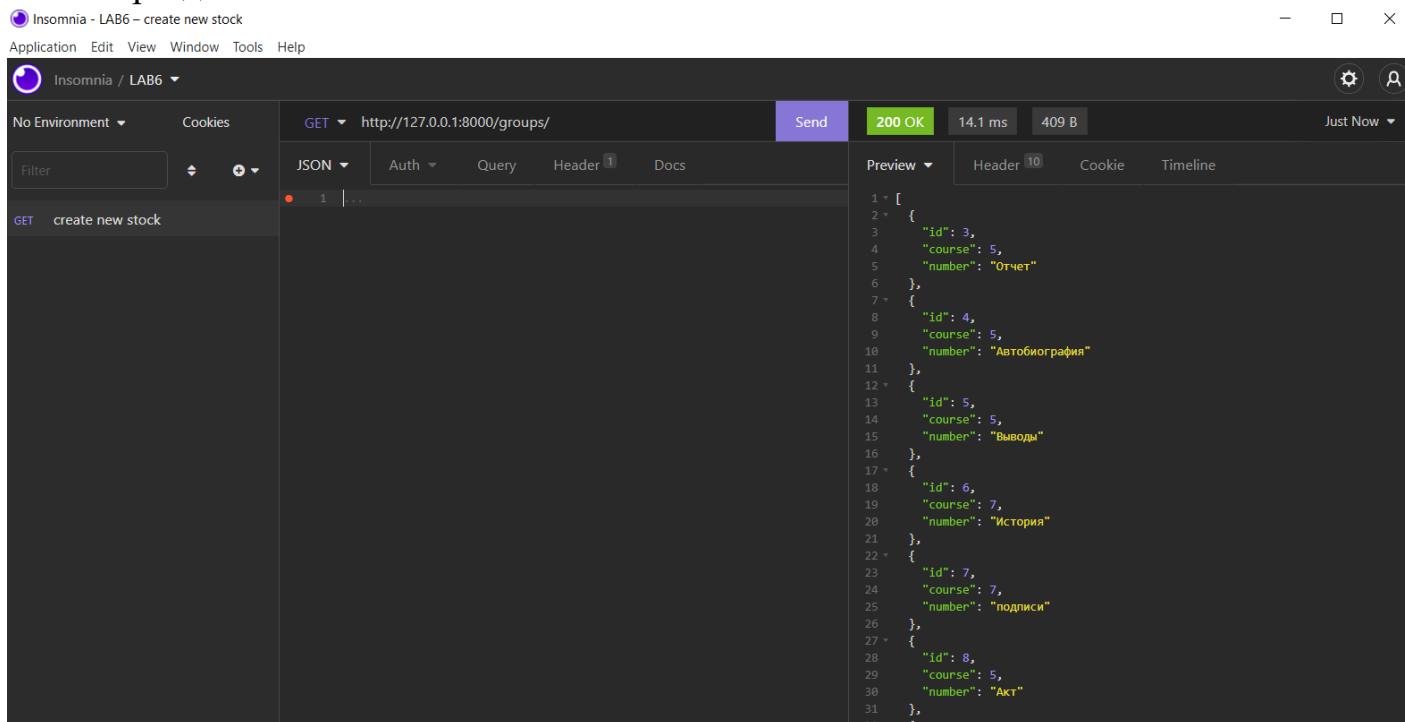
Изменение раздела



Удаление раздела



Список разделов



Просмотр данных через браузер

Список документов

Список документов

[1-й](#)
[2-й](#)
[3-й](#)
[4-й](#)
[4](#)
[6-й](#)
[5](#)
[4](#)

Рубин Р. О. ИУ5Ц-72Б

Информация о разделе

1-й

Номер документа	1-й
-----------------	-----

[Назад](#)

Рубин Р. О. ИУ5Ц-72Б

Список разделов

Список разделов

[Отчет](#)
[Автобиография](#)
[Выводы](#)
[История](#)
[подписи](#)
[Акт](#)
[Начало](#)
[Начало](#)
[Договор](#)

Рубин Р. О. ИУ5Ц-72Б

Информация о документе. Соединение двух таблиц.

Автобиография

Документ	3-й
Раздел	Автобиография

[Назад](#)

Рубин Р. О. ПУСЦ-72Б