МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»

(БГТУ им. В.Г.Шухова)

**Лабораторная работа №3**

дисциплина «Технологии web-программирования»

тема: «**Серверное программирование**»

Выполнил: студент группы ВТ-41

Ковалёв И. Д.

Проверил:

Картамышев С. В.

Белгород 2020

**Цель**: познакомиться с основами backend разработки web-приложений. Познакомиться с основами работы docker. Научиться разворачивать проект, производить его настройку. Научится работать с API в приложении Postman.

**Задание к лабораторной работе:**

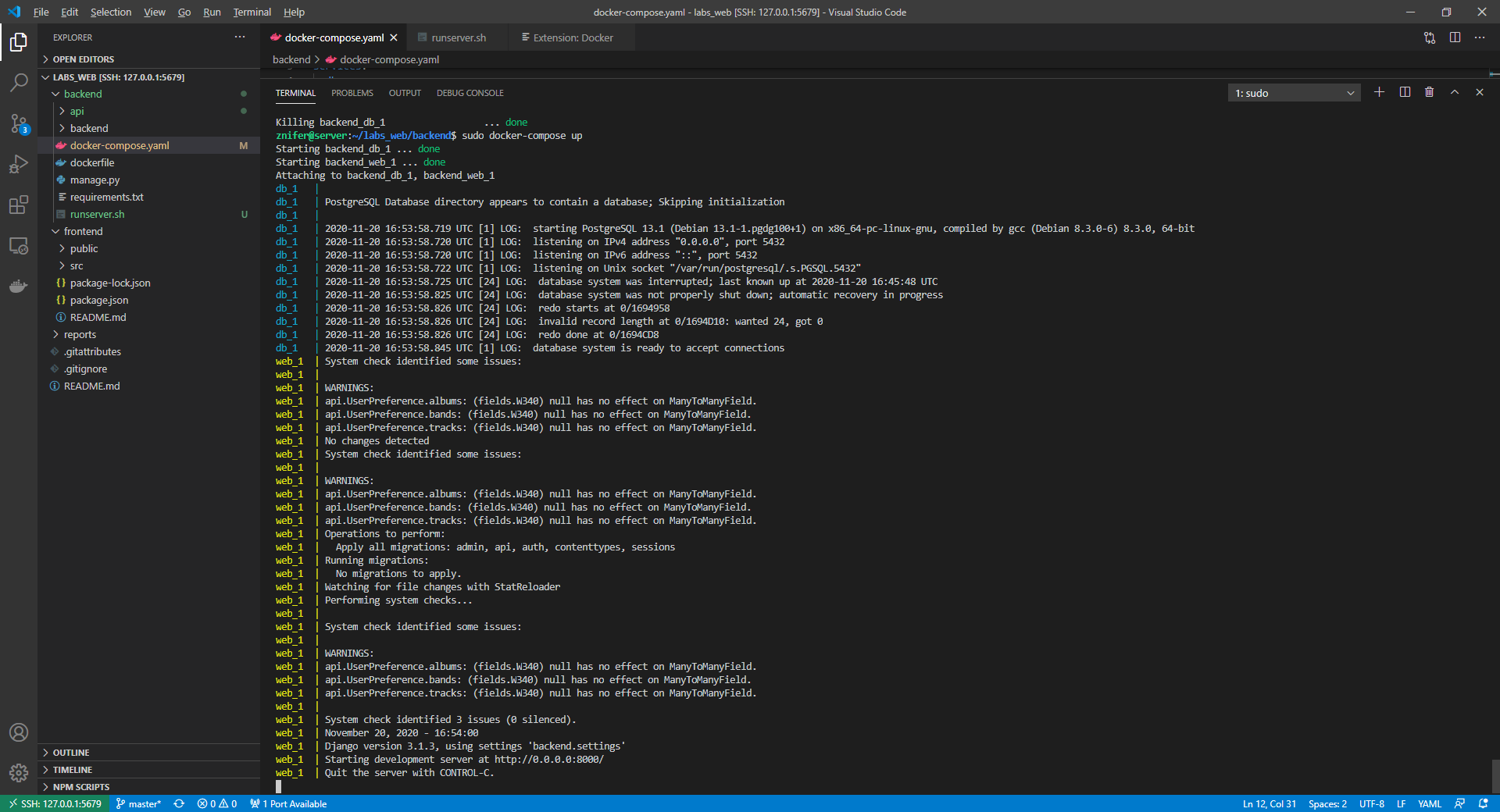
1. Развернуть базовое приложение.
2. Настроить конфигурацию работы приложения с docker.
3. Добавить модуль для работы с API.
4. Добавить несколько контроллеров со статическими данными.
5. Продемонстрировать работу API в Postman.

**Ход работы**

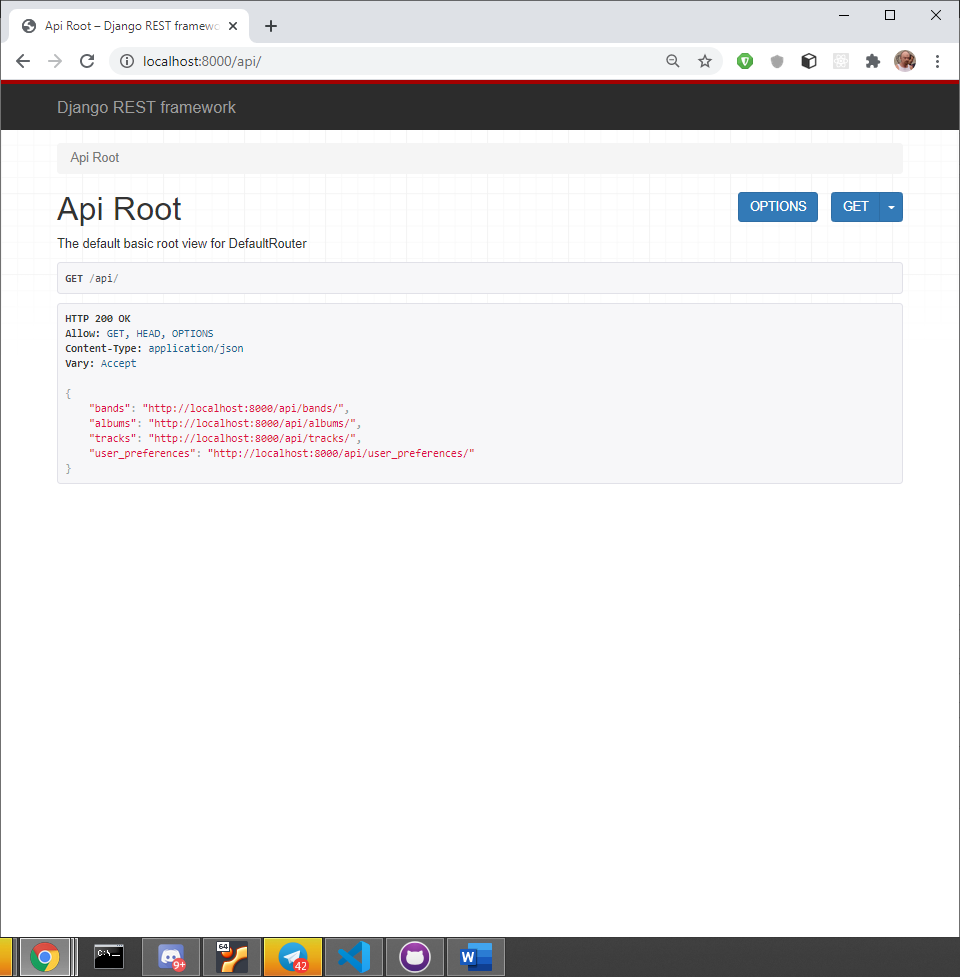
Для реализации серверной части приложения был использован Python и фреймворки Django и Django-rest-framework.

Было развернуто базовое приложение, а также настроены переменные окружения для работы с docker-compose и реализовано API.

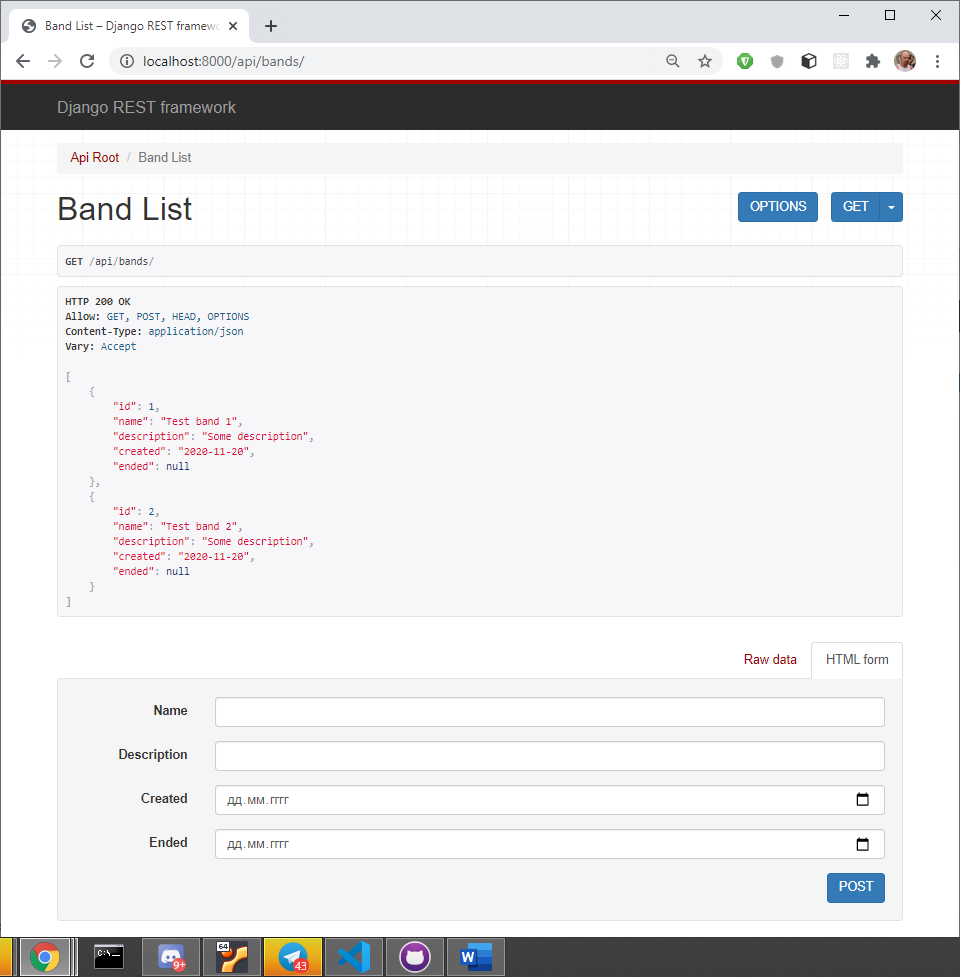
Результат запуска docker-compose:



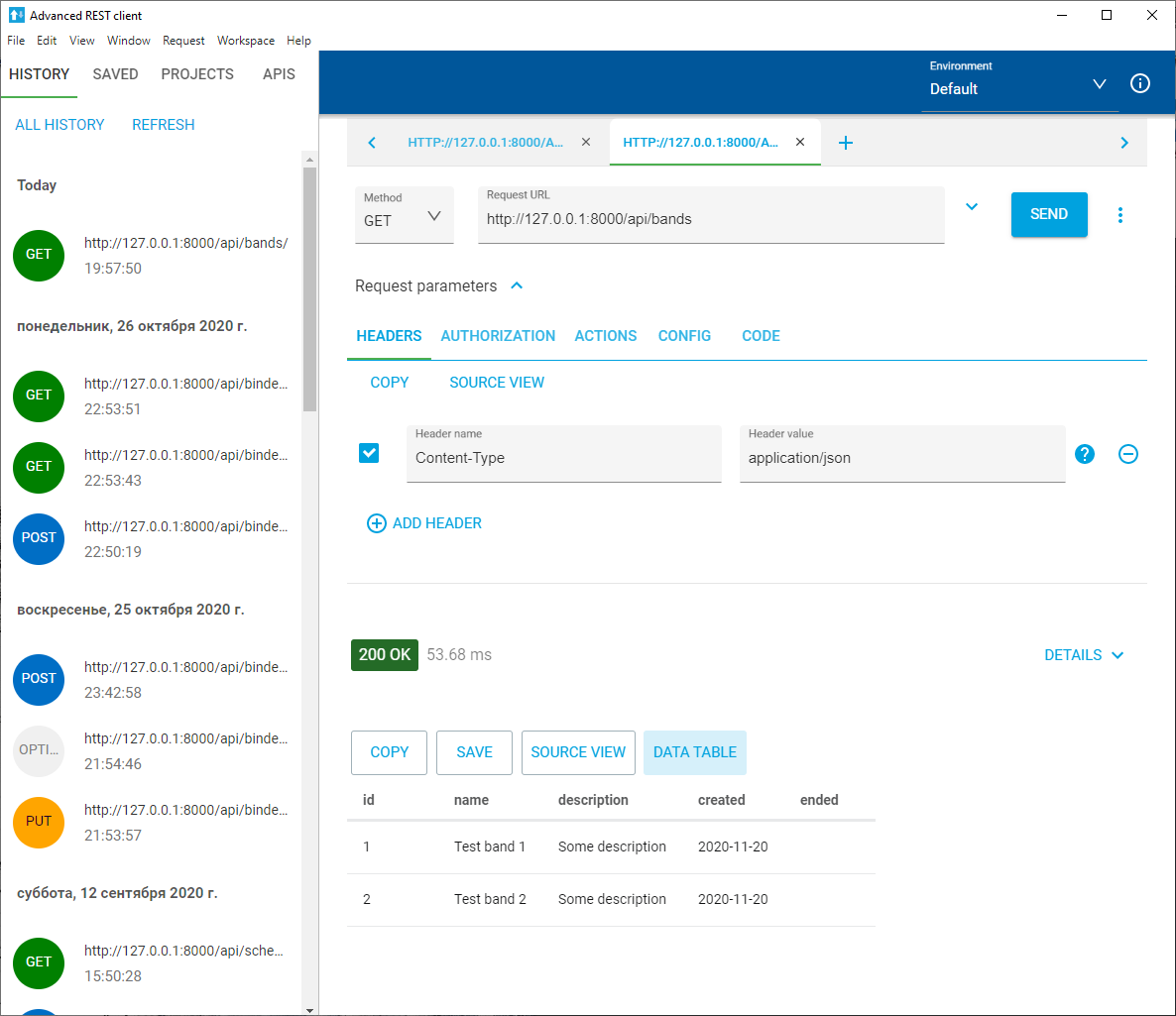
Проверим, действительно ли поднялся сервер:



Поднялся. Добавим некоторые тестовые данные:



И попробуем получить их с помощью Advanced REST client:



Как видно из скриншота, все получилось.

Содержимое dockerfile:

FROM python:3

ENV PYTHONUNBUFFERED=1

WORKDIR /code

COPY requirements.txt /code/

RUN pip install -r requirements.txt

COPY . /code/

Содержимое docker-compose.yaml:

version: "3.8"

services:

  db:

    image: postgres

    environment:

      - POSTGRES\_DB=postgres

      - POSTGRES\_USER=postgres

      - POSTGRES\_PASSWORD=postgres

  web:

    build: .

    command: sh ./runserver.sh

    volumes:

      - .:/code

    ports:

      - "8000:8000"

    depends\_on:

      - db

Скрипт runserver.sh, выполняющий миграции на сервере и запускающий его:

python manage.py makemigrations

python manage.py migrate

python manage.py runserver 0.0.0.0:8000

Содержимое файла settings.py, конфигурирующий работу серверного приложения:

import os

from datetime import timedelta

# Build paths inside the project like this: os.path.join(BASE\_DIR, ...)

BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'r091bbiu##n6%b0n)a3o3rbwtti^0\*h7uqw7!@2nom4dk%f(&e'

# 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

    'rest\_framework',

    #CORS

    'corsheaders',

    'api.apps.ApiConfig',

    'django\_extensions',

    'django\_filters',

]

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 = 'backend.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [],

        '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 = 'backend.wsgi.application'

SIMPLE\_JWT = {

    'ACCESS\_TOKEN\_LIFETIME': timedelta(minutes=5),

    'REFRESH\_TOKEN\_LIFETIME': timedelta(days=1),

    'ROTATE\_REFRESH\_TOKENS': False,

    'BLACKLIST\_AFTER\_ROTATION': True,

    'ALGORITHM': 'HS256',

    'SIGNING\_KEY': SECRET\_KEY,

    'VERIFYING\_KEY': None,

    'AUDIENCE': None,

    'ISSUER': None,

    'AUTH\_HEADER\_TYPES': ('Bearer',),

    'USER\_ID\_FIELD': 'id',

    'USER\_ID\_CLAIM': 'user\_id',

    'AUTH\_TOKEN\_CLASSES': ('rest\_framework\_simplejwt.tokens.AccessToken',),

    'TOKEN\_TYPE\_CLAIM': 'token\_type',

    'JTI\_CLAIM': 'jti',

    'SLIDING\_TOKEN\_REFRESH\_EXP\_CLAIM': 'refresh\_exp',

    'SLIDING\_TOKEN\_LIFETIME': timedelta(minutes=60),

    'SLIDING\_TOKEN\_REFRESH\_LIFETIME': timedelta(days=1),

}

# Database

# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql',

        'NAME': 'postgres',

        'USER': 'postgres',

        'PASSWORD': 'postgres',

        'HOST': 'db',

        'PORT': 5432,

    }

}

# Password validation

# https://docs.djangoproject.com/en/3.0/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/3.0/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC\_URL = '/static/'