МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.Г.ШУХОВА»
(БГТУ им. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Дисциплина: Технологии web-программирования

Лабораторная работа №3.

«Серверное программирование. Laravel»

Выполнил: ст. гр. ПВ-42

Ковынев В.В.

Проверил: Картамышев С.В.

2021

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

Задание к лабораторной работе:

1. Развернуть базовое приложение ASP.NET.
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.

Выполнение:

Для работы был выбран фреймворк ASP.NET, язык программирования C#. В ходе работы был создан контроллер пользователей.

Код контроллера со статическими данными:

```
using Microsoft.AspNetCore.Mvc;

// For more information on enabling Web API for empty projects, visit
https://go.microsoft.com/fwlink/?LinkID=397860

namespace WebBackend.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class StaticDataController : ControllerBase
    {
        private List<string> values = new List<string> { "value1", "value2" };
        // GET: api/<StaticDataController>
        [HttpGet]
        public IEnumerable<string> Get()
        {
            return values;
        }

        // GET api/<StaticDataController>/5
        [HttpGet("{id}")]
        public async Task<ActionResult<string>> Get(int id)
        {
            return Ok(values[id]);
        }

        // POST api/<StaticDataController>
        [HttpPost]
        public async Task<ActionResult> Post([FromBody] string value)
        {
            values.Add(value);
            return Ok();
        }

        // PUT api/<StaticDataController>/5
        [HttpPut("{id}")]
        public async Task<ActionResult> Put(int id, [FromBody] string value)
```

```
        {
            if(id >= values.Count)
            {
                return BadRequest();
            }

            values[id] = value;
            return Ok();
        }

        // DELETE api/<StaticDataController>/5
        [HttpDelete("{id}")]
        public async Task<ActionResult> Delete(int id)
        {
            if (id >= values.Count)
            {
                return BadRequest();
            }

            values.RemoveAt(id);
            return Ok();
        }
    }
}
```

Dockerfile для приложения WebBackend:

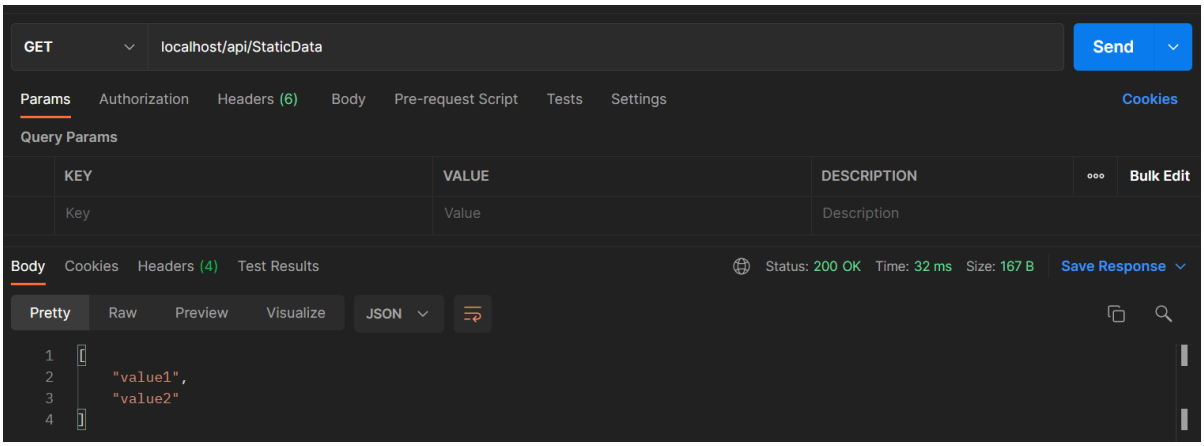
```
FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build
WORKDIR /src
COPY ["WebBackend.csproj", "."]
RUN dotnet restore "./WebBackend.csproj"
COPY . .
WORKDIR "/src/."
RUN dotnet build "WebBackend.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "WebBackend.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "WebBackend.dll"]
```

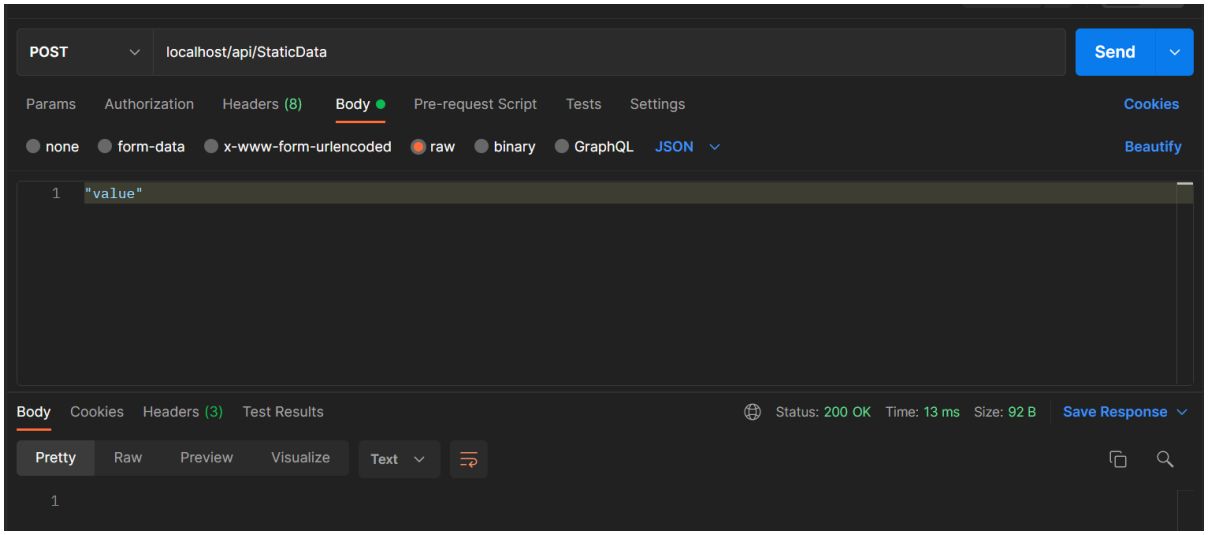
Файл docker-compose.yml:

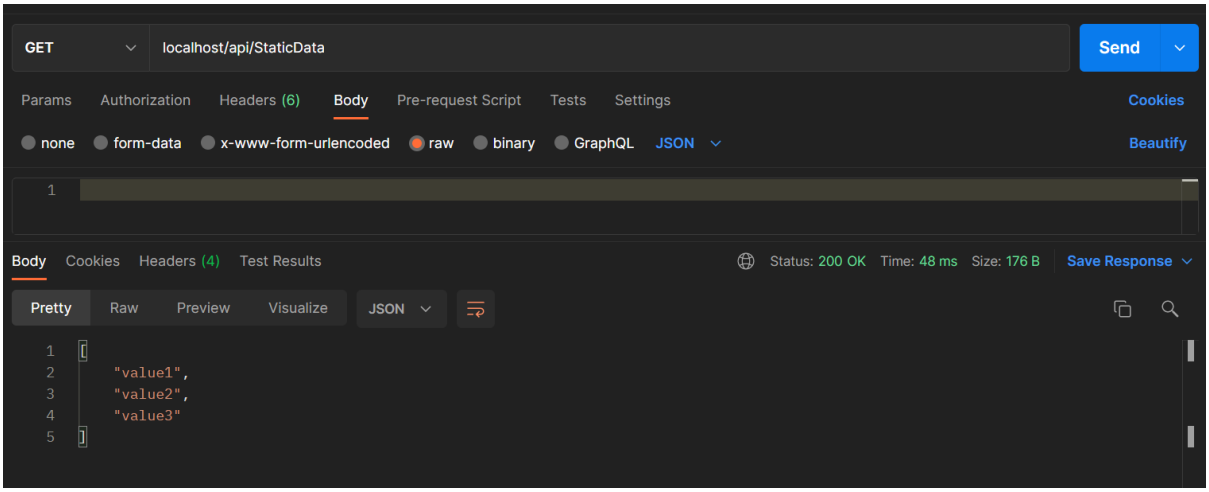
```
version: "3.9"

services:
  backend:
    build: ./WebBackend
    restart: unless-stopped
    depends_on:
      - mysql
    ports:
      - "80:80"
    networks:
      - web-app-network

networks:
```

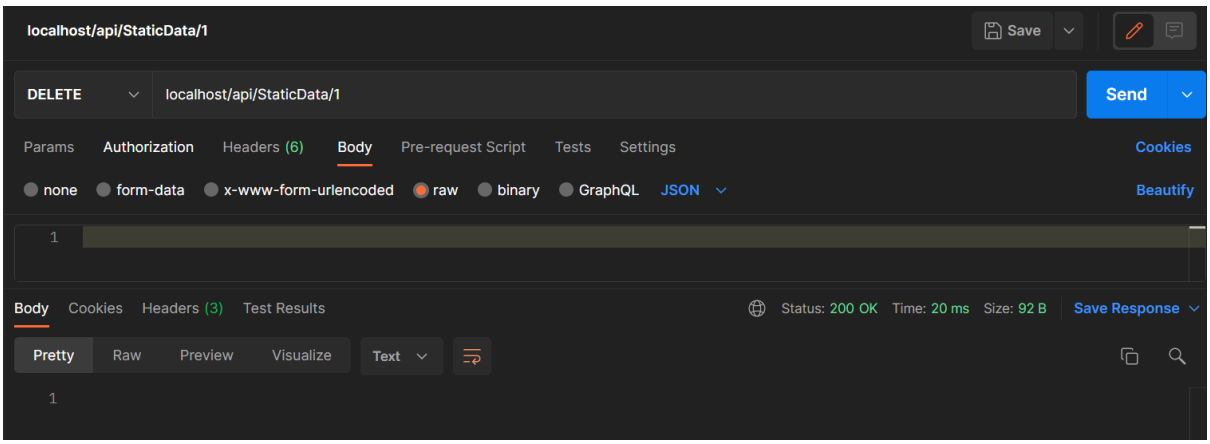
```
web-app-network:
  driver: bridge
```

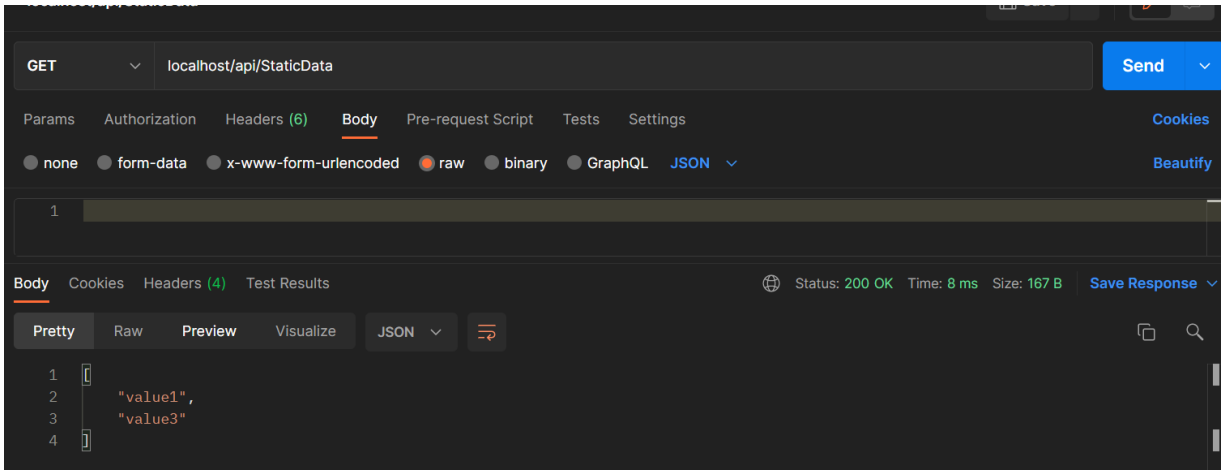
Протестируем работу бэкенда с помощью Postman:








localhost/api/StaticData/1
Save
DELETE
localhost/api/StaticData/1
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
1
Body
Cookies
Headers (3)
Test Results
Status: 200 OK
Time: 20 ms
Size: 92 B
Save Response
Pretty
Raw
Preview
Visualize
Text
1


GET
localhost/api/StaticData
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
1
Body
Cookies
Headers (4)
Test Results
Status: 200 OK
Time: 8 ms
Size: 167 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
1 [
2 "value1",
3 "value3"
4]