

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
Борисенко Валерий Александрович
Проверил: Картамышев С.В.

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. Примеры кода получения пользователей:

Код контроллера со статическими данными:

```
using Microsoft.AspNetCore.Mvc;

namespace BackendApp.Controllers;

[ApiController]
[Route("[controller]")]
public class WeatherForecastController : ControllerBase
{
    private static readonly string[] Summaries = new[]
    {
        "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy", "Hot",
        "Sweltering", "Scorching"
    };

    private readonly ILogger<WeatherForecastController> _logger;

    public WeatherForecastController(ILogger<WeatherForecastController> logger)
    {
        _logger = logger;
    }
}
```

```

    }

    [HttpGet(Name = "GetWeatherForecast")]
    public IEnumerable<WeatherForecast> Get()
    {
        return Enumerable.Range(1, 5).Select(index => new WeatherForecast
        {
            Date = DateTime.Now.AddDays(index),
            TemperatureC = Random.Shared.Next(-20, 55),
            Summary = Summaries[Random.Shared.Next(Summaries.Length)]
        })
        .ToArray();
    }
}

```

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 ["BackendApp.csproj", "."]
RUN dotnet restore "BackendApp.csproj"
COPY . .
WORKDIR "/src/"
RUN dotnet build "BackendApp.csproj" -c Release -o /app/build

```

```

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

```

```

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

```

Файл docker-compose.yml:

```
version: "3.9"
```

```
services:
```

```
  database:
```

```
    container_name: database
```

```
    image: postgres:14
```

restart: unless-stopped
environment:
 POSTGRES_PASSWORD: postgrespassword
 POSTGRES_USER: boosterjz
 POSTGRES_DB: database
ports:
 - "5432:5432"
networks:
 - web-network
volumes:
 - dbdata:/var/lib/postgresql/data

backend:
 container_name: backend
 build: ./BackendApp
 ports:
 - "8080:80"
 restart: unless-stopped
 depends_on:
 - db
 networks:
 - web-network

frontend:
 container_name: frontend
 build: .
 ports:
 - "80:80"
 restart: unless-stopped
 depends_on:
 - backend
 volumes:
 - ./nginx:/etc/nginx/conf.d
 networks:
 - web-network

networks:
 web-network:
 driver: bridge

volumes:
 dbdata:
 driver: local

http://localhost:8080/WeatherForecast

Save

GET

http://localhost:8080/WeatherForecast

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION		Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 167 ms

Size: 651 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    {
3      "date": "2022-01-20T10:36:04.3353235+00:00",
4      "temperatureC": 38,
5      "temperatureF": 100,
6      "summary": "Freezing"
7    },
8    {
9      "date": "2022-01-21T10:36:04.3382883+00:00",
10     "temperatureC": -14,
11     "temperatureF": 7,
12     "summary": "Scorching"
13   },
14   {
15     "date": "2022-01-22T10:36:04.338305+00:00",
16     "temperatureC": 2,
17     "temperatureF": 35,
18     "summary": "Mild"
```