

**федеральное государственное автономное образовательное  
учреждение высшего образования**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)  
(Факультет информационных технологий)**

*(Институт Принтмедиа и информационных технологий)  
Кафедра Информатики и информационных технологий*

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**Дисциплина: BackEnd-разработка**

**Тема: Создание приложения на основе класса WebApplication на основе ASP.NET  
Core**

**Выполнил(а): студент(ка) группы 221-3711**

Ежов Тимофей Алексеевич  
(Фамилия И.О.)

**Дата, подпись 13.02.2024**

**Проверил:** \_\_\_\_\_  
(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_  
(Дата) (Подпись)

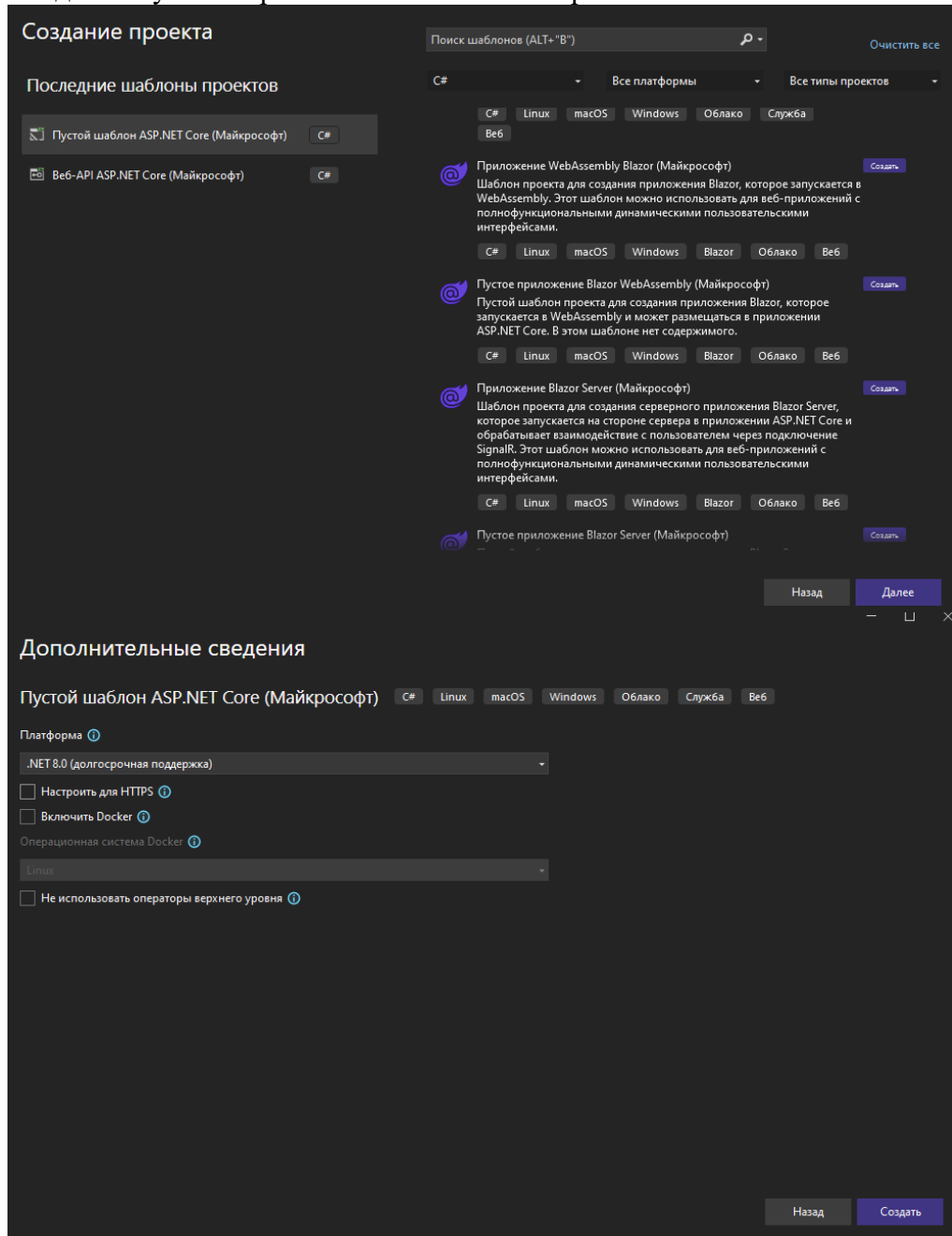
**Замечания:** \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

**Москва 2024**

## Этапы создания:

- 1) Создание пустого проекта из шаблона веб приложение ASP.Net core в Visual Studio



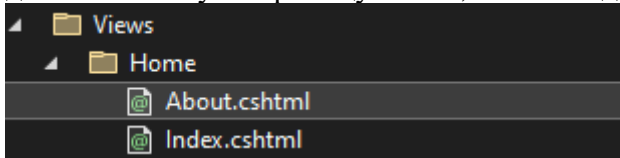
- 2) Создаём в Programm.cs экземпляр WebApplication.Builder, используя WebApplication.CreateBuilder(args).

```
var builder = WebApplication.CreateBuilder(args);  
  
var LAB1 = builder.Build();
```

- 3) Настройка маршрутизации.

```
using Microsoft.AspNetCore.Mvc;
namespace LAB_1_True.Controllers
{
    Ссылка: 0
    public class HomeController : Controller
    {
        Ссылка: 0
        public IActionResult Index() { return View(); }
        Ссылка: 0
        public IActionResult About() { return View(); }
    }
}
```

- 4) Далее создадим методы для вызова двух представлений, домашнюю страницу Index и дополнительную страницу About, затем создаем папку Views и в ней Home.



- 5) Изменим файлы представления страниц и добавим гиперссылки в них друг на друга

```
@{
    ViewData["Title"] = "Main";
}
<h2> Лабораторная работа 1</h2>
<p>Главная страница</p>
<a href="/Home/about">Перейти на страницу About</a>
```

```
@{
    ViewData["Title"] = "About";
}
<h2> Лабораторная работа 1</h2>
<p>Создаём новый проект ASP.NET Core, с использованием класса WebApplication.</p>
<a href="/Home/">Вернуться в главное</a>
```

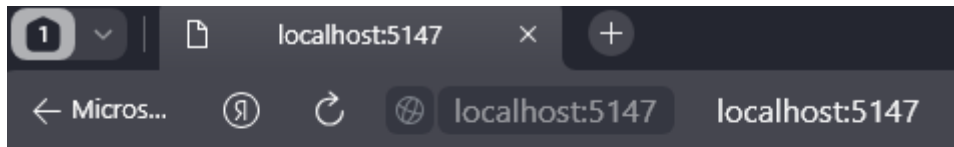
- 6) Настройка маршрутизации с Programm.cs

```
LAB1.UseRouting();
LAB1.MapControllerRoute(name: "default", pattern: "{controller=Home}/{action=Index}");
```

используем стандартный шаблон.



- 7) Запуск приложения. Используем метод Run  
После чего открывается окно браузера с адресом localhost:номер порта/



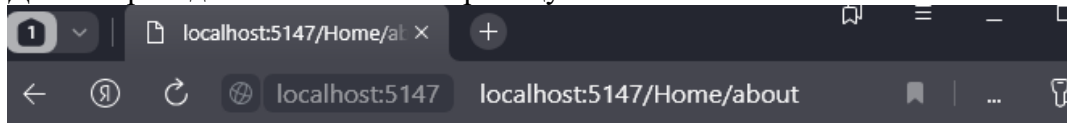
## Лабораторная работа 1

Главная страница

[Перейти на страницу About](#)

Главная страница

Далее переходим по ссылке на страницу About



## Лабораторная работа 1

Создаём новый проект ASP.NET Core, с использованием класса WebApplication.

[Вернуться в главное](#)

Всё работает, переход обратно также работает.

### Изменение по сравнению с базовым шаблоном:

Написанный код основан на базовом шаблоне ASP.NET Core, однако были добавлены дополнительные настройки маршрутизации с использованием MVC, создан контроллер и два представления с возможностью перехода между ними.

### Листинг

Programm.cs

```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllersWithViews();
var LAB1 = builder.Build();

LAB1.UseRouting();
LAB1.MapControllerRoute(name: "default", pattern:
"{controller=Home}/{action=Index}");

LAB1.Run();
```

HomeController.cs

```
using Microsoft.AspNetCore.Mvc;
namespace LAB_1_True.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index() { return View(); }
        public IActionResult About() { return View(); }
    }
}
```

Главная страница

```
@{
    ViewData["Title"] = "Main";
}
<h2> Лабораторная работа 1</h2>
<p>Главная страница</p>
<a href="/Home/about">Перейти на страницу About</a>
```

Дополнительная страница

```
@{
    ViewData["Title"] = "About";
}
<h2> Лабораторная работа 1</h2>
<p>Создаём новый проект ASP.NET Core, с использованием класса WebApplication.</p>
<a href="/Home/">Вернуться в главное</a>
```