



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки
09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 8

Дисциплина: Backend разработка

Тема:

Выполнил(а): студент(ка) группы 231-336

Канищев И.М.
(Фамилия И.О.)

Дата, подпись (Дата) (Подпись)

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

Дата, подпись (Дата) (Подпись)

Замечания:

Москва
2025

1. Работа с состояниями на сервере (Session)

- **Сессия** хранится на сервере и позволяет сохранять данные для конкретного пользователя между запросами.
- Настройка сессии в Program.cs:

```
using Microsoft.AspNetCore.Builder;
using Microsoft.Extensions.DependencyInjection;

var builder = WebApplication.CreateBuilder(args);

// Подключаем MVC
builder.Services.AddControllersWithViews();

// Подключаем кэш для сессий и сессии
builder.Services.AddDistributedMemoryCache();
builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromMinutes(30);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
});

var app = builder.Build();

app.UseDefaultFiles();
app.UseStaticFiles();
app.UseRouting();
app.UseSession();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Session}/{action=Index}/{id?}");

app.Run();
```

Пример использования сессии в контроллере:

```
using Microsoft.AspNetCore.Mvc;

public class SessionController : Controller
{
    public IActionResult SetSession()
    {
        HttpContext.Session.SetString("Username", "Alice");
        return Content("Session value set!");
    }

    public IActionResult GetSession()
    {
        var username = HttpContext.Session.GetString("Username") ?? "No value";
        return Content($"Session value: {username}");
    }
}
```

- SetString сохраняет данные в сессии.

- GetString извлекает данные.
- Сессии уникальны для каждого пользователя.

2. Работа с состояниями на клиенте

2.1 Cookies

- Cookies хранятся в браузере и отправляются на сервер при каждом запросе.
- Пример установки и чтения cookie в контроллере:

```
public IActionResult SetCookie()
{
    Response.Cookies.Append("UserToken", "123456", new CookieOptions
    {
        Expires = DateTimeOffset.Now.AddMinutes(60)
    });
    return Content("Cookie set!");
}

public IActionResult GetCookie()
{
    var value = Request.Cookies["UserToken"] ?? "No cookie";
    return Content($"Cookie value: {value}");
}
```

2.2 localStorage и sessionStorage

- Данные хранятся на клиенте и доступны через JavaScript.
- **localStorage** — хранит данные постоянно.
- **sessionStorage** — хранит данные только для текущей вкладки браузера.

Пример использования на странице HTML:

```
<script>
    // localStorage
    localStorage.setItem("username", "Alice");
    console.log("localStorage username:", localStorage.getItem("username"));

    // sessionStorage
    sessionStorage.setItem("sessionId", "ABC123");
    console.log("sessionStorage sessionId:",
sessionStorage.getItem("sessionId"));
</script>
```

3. Пример передачи данных между клиентом и сервером

- **Сессии:** сервер сохраняет данные → клиент делает запрос → сервер возвращает сохранённое значение.
- **Cookies:** сервер устанавливает cookie → клиент автоматически отправляет cookie при последующих запросах.
- **localStorage/sessionStorage:** данные хранятся на клиенте → клиент может отправить данные на сервер через AJAX.

Пример AJAX-запроса с localStorage:

```
<script>
  fetch("/api/echo", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ username: localStorage.getItem("username") })
  })
  .then(res => res.text())
  .then(data => console.log(data));
</script>
```

4. Тестирование приложения

Работа с состояниями в ASP.NET Core

localStorage и sessionStorage

Тестирование сессии и cookies

- [Set Session](#)
- [Get Session](#)
- [Set Cookie](#)
- [Get Cookie](#)

Работа с состояниями в ASP.NET Core

localStorage и sessionStorage

[Set Storage](#) [Get Storage](#)

Тестирование сессии и cookies

- [Set Session](#)
- [Get Session](#)
- [Set Cookie](#)
- [Get Cookie](#)

localhost:7190

Storage values set!

OK

Работа с состояниями в ASP.NET Core

localStorage и sessionStorage

[Set Storage](#) [Get Storage](#)

Тестирование сессии и cookies

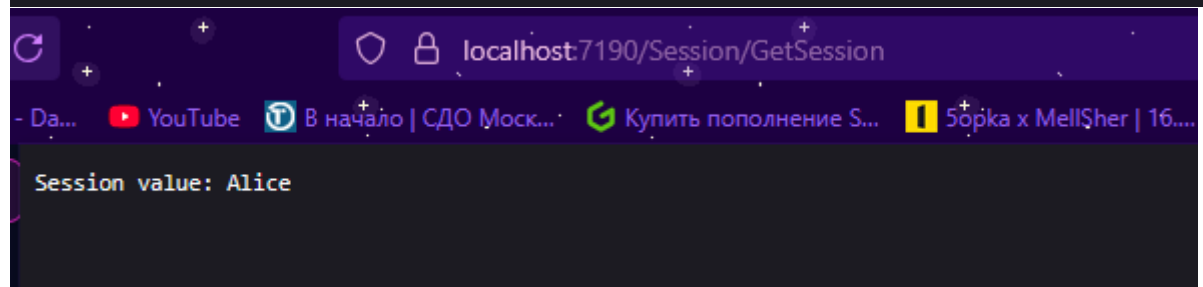
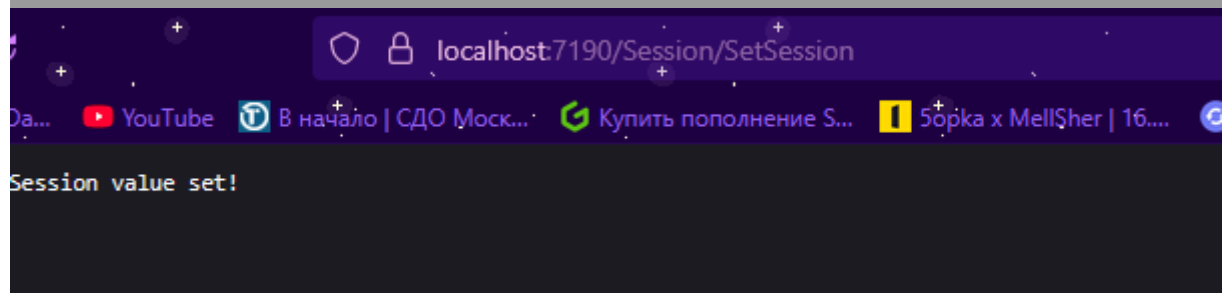
- [Set Session](#)
- [Get Session](#)
- [Set Cookie](#)
- [Get Cookie](#)

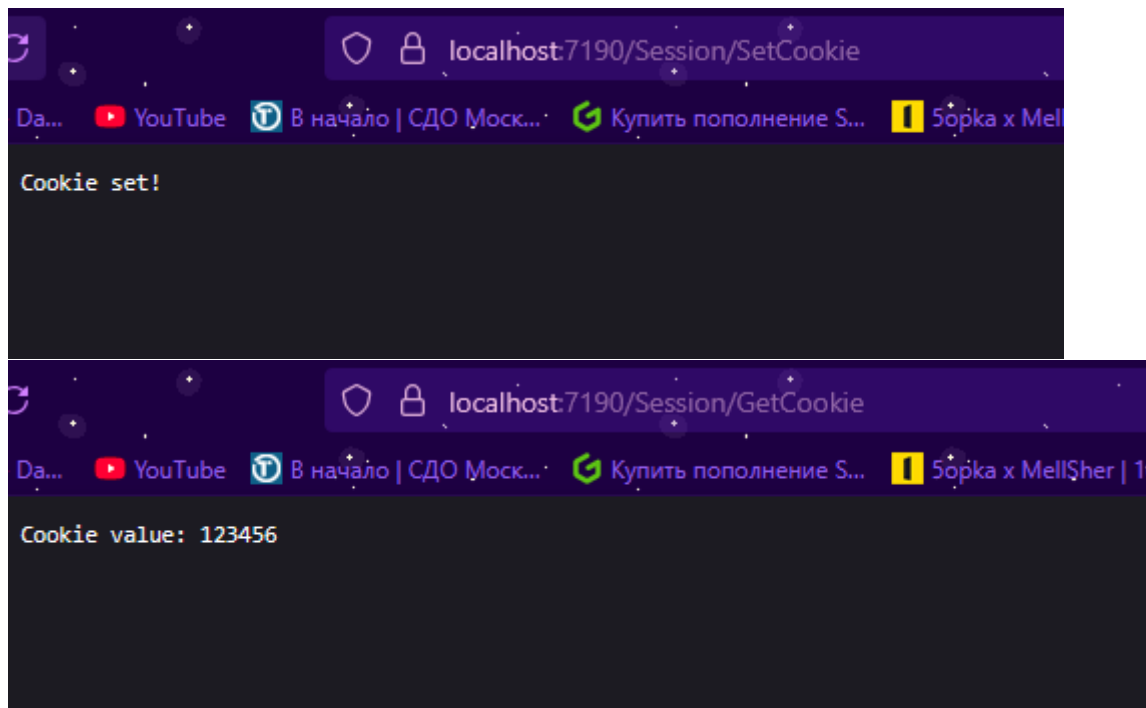
localhost:7190

localStorage username: Alice
sessionStorage sessionId: ABC123

☐ Не позволять localhost:7190 снова вас запрашивать

OK





- **Сессии:** безопасны, хранят данные на сервере, подходят для чувствительной информации.
- **Cookies:** данные хранятся на клиенте, подходят для простых настроек и идентификаторов.
- **localStorage / sessionStorage:** удобны для хранения данных на клиенте без постоянного обращения к серверу.
- Комбинированное использование позволяет эффективно управлять состояниями и оптимизировать работу приложения.