

State Management Lab

Lab 1: Add a Cookie

Open **HomeController.cs** file and locate the **Index()** method and modify the code in bold shown below.

```
[HttpGet]
public IActionResult Index()
{
    // Has the user been here before?
    ViewData["Returning"] = (Request.Cookies["Returning"] != null);
    // If the user has not been there, send a cookie out
    if(!Convert.ToBoolean(ViewData["Returning"])) {
        CookieOptions options = new()
        {
            Expires = DateTime.Now.AddMinutes(1)
        };
        Response.Cookies.Append("Returning", "true", options);
    }

    return View();
}
```

Open the **\Views\Home\Index.cshtml** file and add the following in between the other two **<div>** tags

```
<div class="text-center">
    @if (Convert.ToBoolean(ViewData["Returning"])) {
        <h3>Welcome back!</h3>
    }
    else {
        <h3>Welcome for the first time!</h3>
    }
</div>
```

Try it Out

Run the application and you should see the text "Welcome for the first time!" appear.

Refresh the page and you should see the text "Welcome Back" appear.

Lab 2: Wrapper Around Session State

Right mouse-click on the **\Models** folder and add a new Class named **UserSession.cs**.

```
namespace AdvWorks.Models
{
    public class UserSession
    {
        public UserSession(IHttpContextAccessor httpAccessor)
        {
            _httpAccessor = httpAccessor;
        }

        protected readonly IHttpContextAccessor _httpAccessor;

        public string LastColor
        {
            get { return
                _httpAccessor.HttpContext.Session.GetString("LastColor"); }
            set { _httpAccessor.HttpContext.Session.SetString("LastColor",
                value); }
        }

        public decimal LastCost
        {
            get { return
                Convert.ToDecimal(_httpAccessor.HttpContext.Session.GetString("LastC
                ost")); }
            set { _httpAccessor.HttpContext.Session.SetString("LastCost",
                value.ToString()); }
        }

        public decimal LastPrice
        {
            get { return
                Convert.ToDecimal(_httpAccessor.HttpContext.Session.GetString("LastP
                rice")); }
            set { _httpAccessor.HttpContext.Session.SetString("LastPrice",
                value.ToString()); }
        }
    }
}
```

Open the **Program.cs** file and add the following lines after the call to the `builder.Services.AddSingleton<AppSettings>()`;

```
// The next three lines are for session state
builder.Services.AddDistributedMemoryCache();
builder.Services.AddSession();
builder.Services.AddHttpContextAccessor();
```

Next, add the following lines immediately after the above lines:

```
// Add a UserSession object
builder.Services.AddScoped<UserSession>();
```

Scroll down and locate the call to **app.UseRouting()**, and add the following line immediately after this call.

```
app.UseSession();
```

Open the **HomeController.cs** file modify the Constructor to inject the *UserSession* object and add a new field.

```
public HomeController(ILogger<HomeController> logger,
                    UserSession session)
{
    _logger = logger;
    _session = session;
}

private readonly UserSession _session;
```

Modify the **Index()** method and add an else condition

```
[HttpGet]
public IActionResult Index()
{
    // Has the user been here before?
    ViewData["Returning"] = (Request.Cookies["Returning"] != null);
    // If the user has not been there, send a cookie out
    if (!Convert.ToBoolean(ViewData["Returning"])) {
        CookieOptions options = new()
        {
            Expires = DateTime.Now.AddMinutes(10)
        };
        Response.Cookies.Append("Returning", "true", options);
    }
    else {
        _session.LastColor = "Blue";
        _session.LastCost = 10;
        _session.LastPrice = 100;
    }

    return View();
}
```

Open the **ProductMaintController.cs** file and modify the constructor to inject the *UserSession* object and add a new field.

```
public ProductMaintController(AppSettings settings,
                             UserSession session)
{
    _settings = settings;
    _session = session;
}

private readonly UserSession _session;
private readonly AppSettings _settings;
```

Modify the ProductDetail() method to look like the following:

```
[HttpGet]
public IActionResult ProductDetail()
{
    ViewBag.Title = "Product Information";
    ProductViewModel vm = new();

    if (string.IsNullOrEmpty(_session.LastColor)) {
        // Retrieve items from configuration settings file
        vm.DefaultCost = _settings.DefaultCost;
        vm.DefaultPrice = _settings.DefaultPrice;
        vm.DefaultColor = _settings.DefaultColor;
    }
    else {
        // Retrieve items from session
        vm.DefaultCost = _session.LastCost;
        vm.DefaultPrice = _session.LastPrice;
        vm.DefaultColor = _session.LastColor;
    }

    return View(vm);
}
```

Try it Out

Run the application and make sure it is the first time in.

If not, Click on the Product Maintenance link, open your F12 browser tools and delete the cookie, then stop the application and rerun the application

Click on the Product Maintenance link, then click on the Add button

You should now see the Color, Cost and Price you entered into the appsettings.json file.

Click on the "Adventure Works" at the top left of the page.

Click on the Product Maintenance link and click on the Add button again

You should now see the Color, Cost and Price you entered into the session class.