

FIT5032 - Internet Applications Development

WEEK 06C - USING SESSIONS & COOKIES

Last updated: 18th August 2018

Author: Jian Liew

Introduction

There is no need to complete this tutorial. It functions as a supplementary material to showcase how to use Sessions & Cookies.

It is quite common to store certain information on the users browsers instead of persisting the information in the database. For example, if you are planning to implement a "Shopping Cart" like feature, it would be a good idea to persist this information in the browser instead. When the user comes back at another time, the items added in the shopping cart will still be present.

Objectives

Estimated Time To Complete - 1 hours

Upon the completion of this tutorial, you will gain a basic understanding of

- What are Sessions & Cookies?
- How to use Sessions & Cookies.

Doubtfire Submission

- **None**

Session

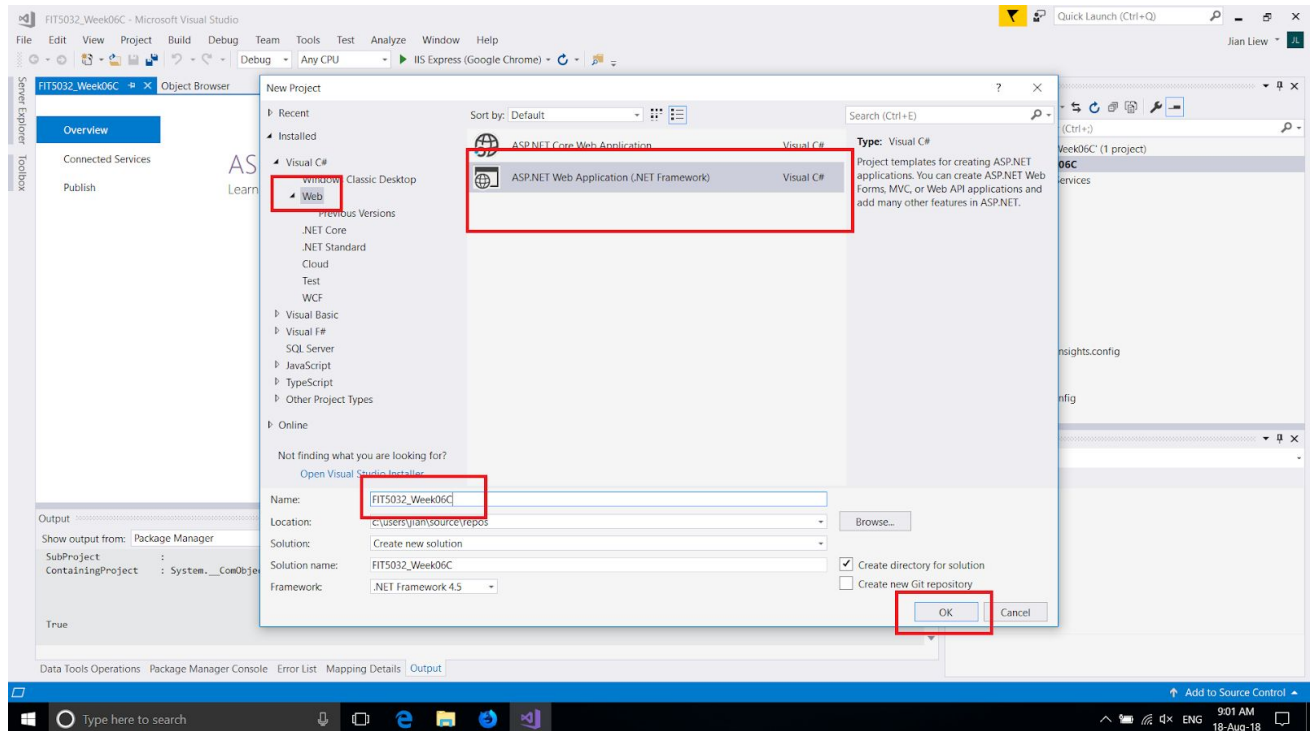
ASP.NET session state enables you to store and retrieve values for a user as the user navigates ASP.NET pages in a Web application. HTTP is a stateless protocol. This means that a Web server treats each HTTP request for a page as an independent request. The server retains no knowledge of variable values that were used during previous requests. **ASP.NET session state identifies requests from the same browser during a limited time window as a session, and provides a way to persist variable values for the duration of that session.** By default, ASP.NET session state is enabled for all ASP.NET applications.

Sessions are very useful when you are planning to maintain the state of certain objects when the user navigates through your website.

Cookies

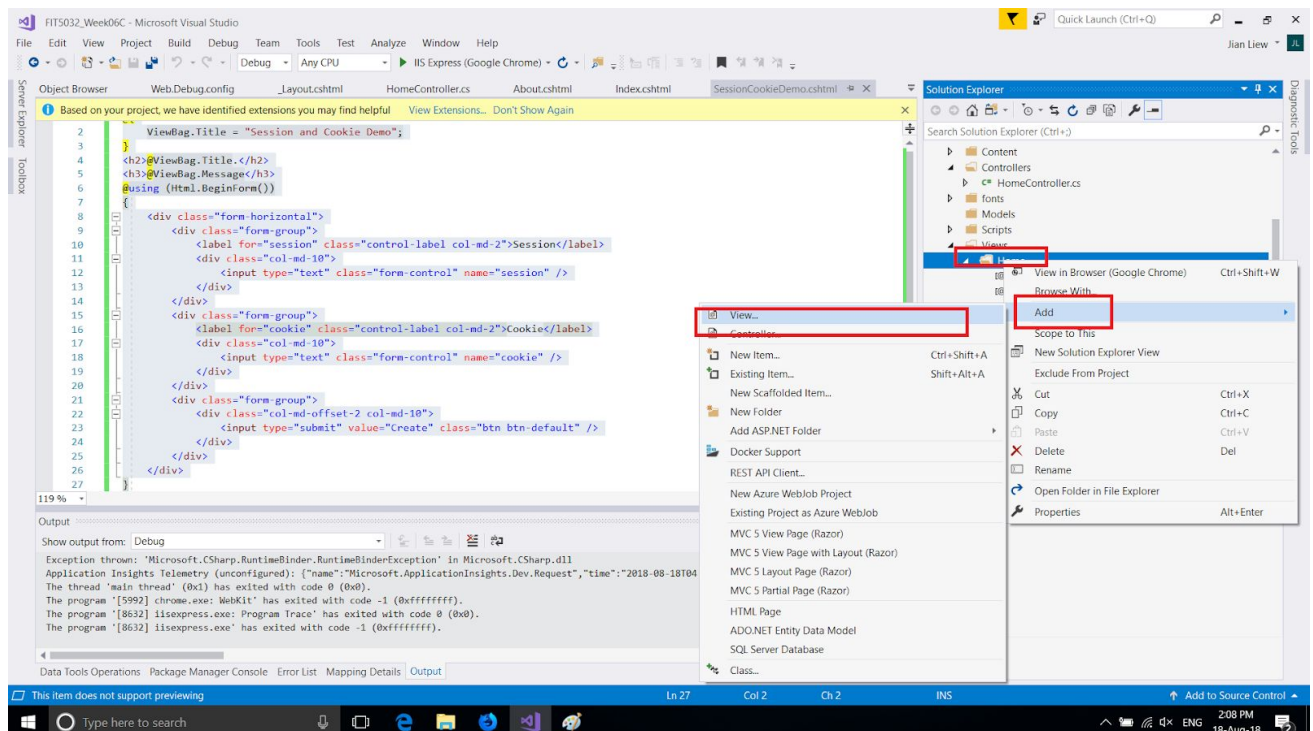
A cookie is a small bit of text that accompanies requests and pages as they go between the Web server and browser. The cookie contains information the Web application can read whenever the user visits the site.

Step 1



Step 2

I will create a new .cshtml file called SessionCookieDemo.cshtml under the Home folder.



Step 3

These are the codes for the SessionCookieDemo.cshtml

```
@{
    ViewBag.Title = "Session and Cookie Demo";
}
<h2>@ViewBag.Title.</h2>
<h3>@ViewBag.Message</h3>
@using (Html.BeginForm())
{
    <div class="form-horizontal">
        <div class="form-group">
            <label for="session" class="control-label col-md-2">Session</label>
            <div class="col-md-10">
                <input type="text" class="form-control" name="session" />
            </div>
        </div>
        <div class="form-group">
            <label for="cookie" class="control-label col-md-2">Cookie</label>
            <div class="col-md-10">
                <input type="text" class="form-control" name="cookie" />
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default" />
            </div>
        </div>
    </div>
}
```

Step 4

You will also need to introduce at the _Layout.cshtml file.

```
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li>@Html.ActionLink("Home", "Index", "Home")</li>
        <li>@Html.ActionLink("About", "About", "Home")</li>
        <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
        <li>@Html.ActionLink("Session & Cookie Demo", "SessionCookieDemo", "Home")</li>
    </ul>
</div>
```

Step 5 - Replace your HomeController.cs with the following codes. **There are no validations in this demo.**

```
using System;
using System.Web;
using System.Web.Mvc;

namespace FIT5032_Week06C.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult SessionCookieDemo()
        {
            ViewBag.Title = "Session and Cookie Demo";
            return View();
        }

        [HttpPost]
        public ActionResult SessionCookieDemo(FormCollection formCollection)
        {
            Session["Demo"] = formCollection["session"];
            HttpCookie myCookie = new HttpCookie("UserSettings");
            myCookie["Cookie"] = formCollection["cookie"];
            myCookie.Expires = DateTime.Now.AddDays(1d);
            Response.Cookies.Add(myCookie);
            return View();
        }

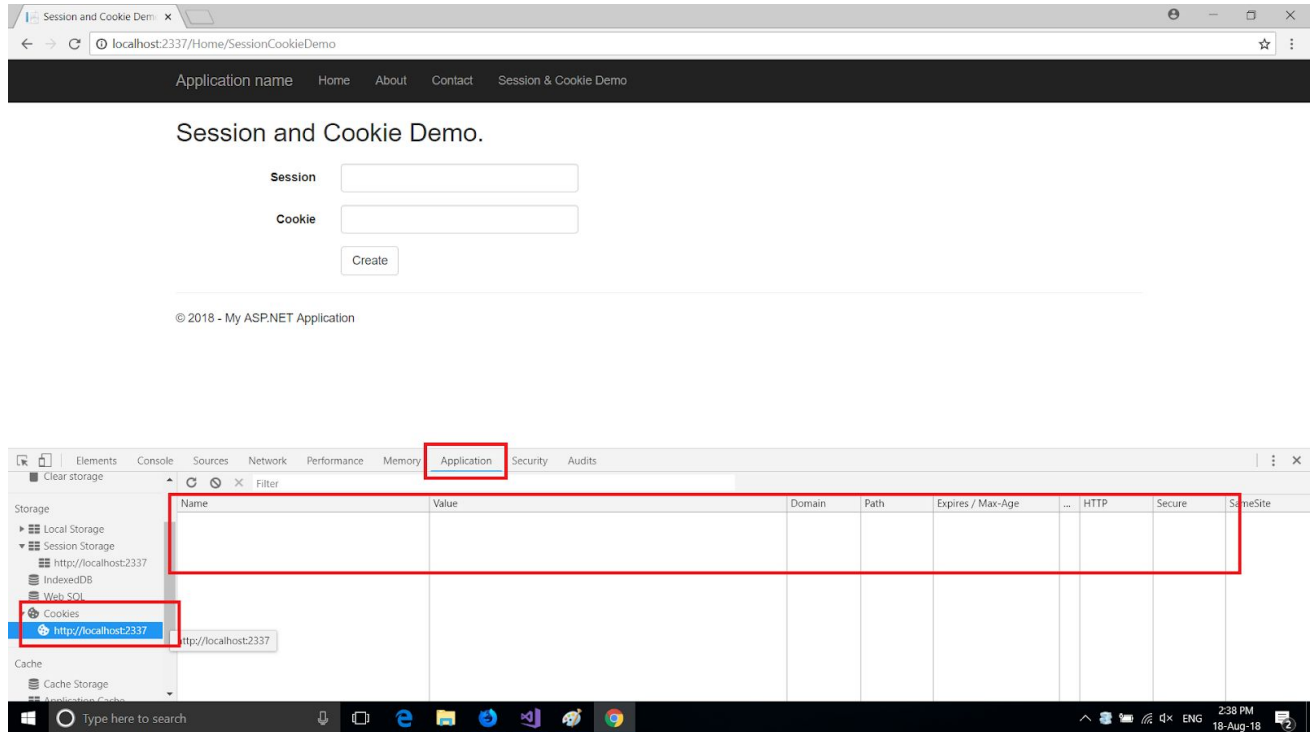
        public ActionResult About()
        {
            ViewBag.Message = Session["Demo"];
            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";
            if (Request.Cookies["UserSettings"] != null)
            {
                var value = Request.Cookies["UserSettings"];
                var storedValue = value["Cookie"];
                ViewBag.Message = "Your cookie value is " + storedValue;
            }

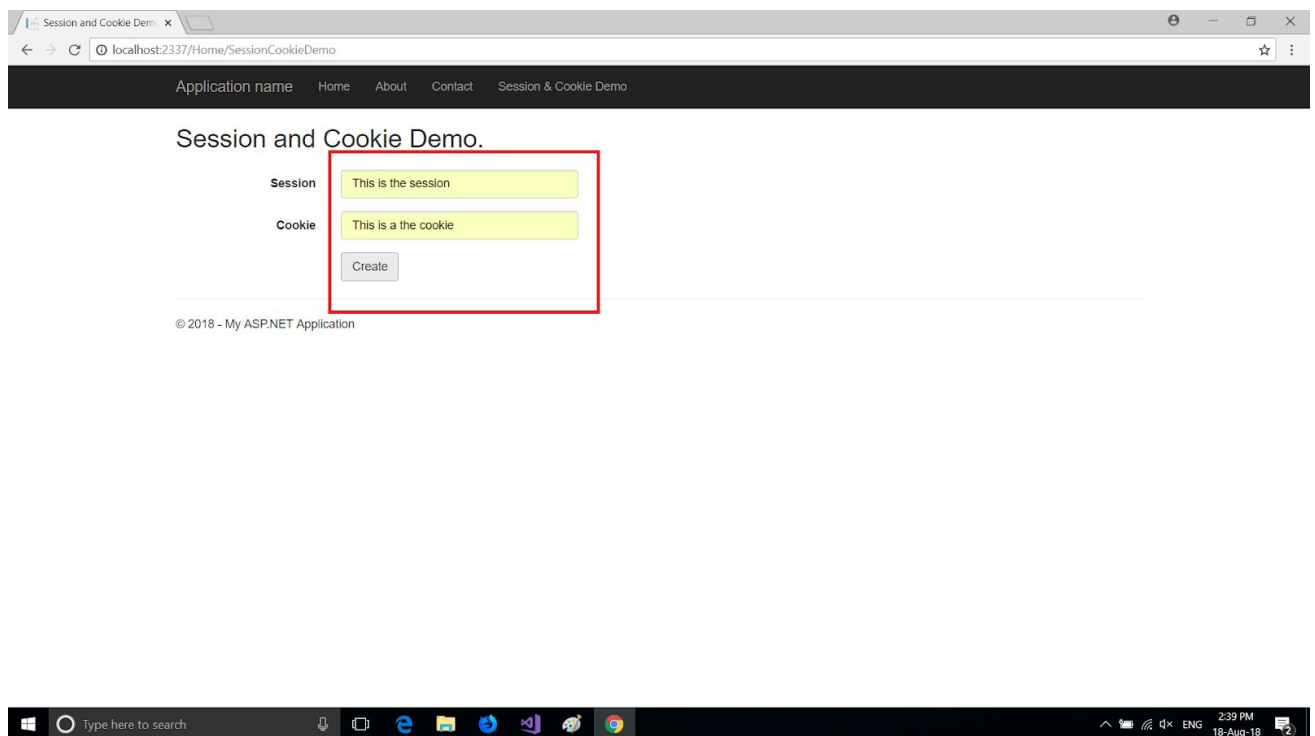
            return View();
        }
    }
}
```

Step 6

We will now run it. In order to see it in action, I will use my developer tools on Google Chrome. (Opened using CTRL + SHIFT + I or F12 or Right Click and then "Inspect").

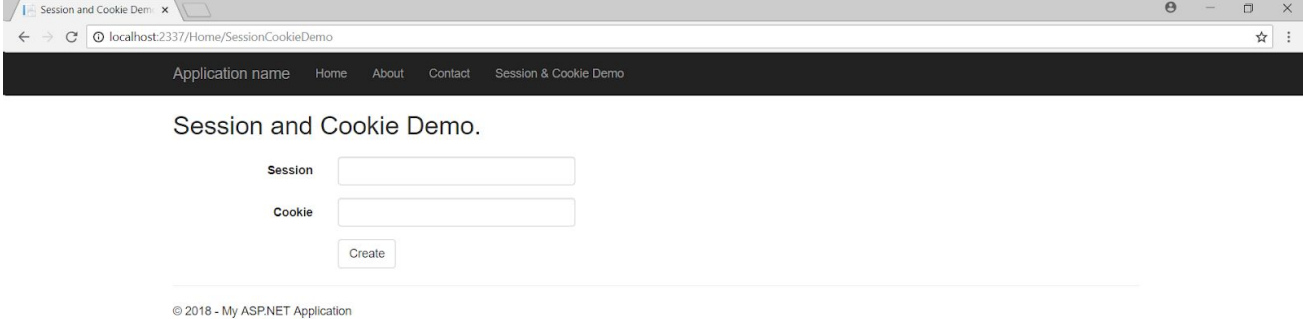
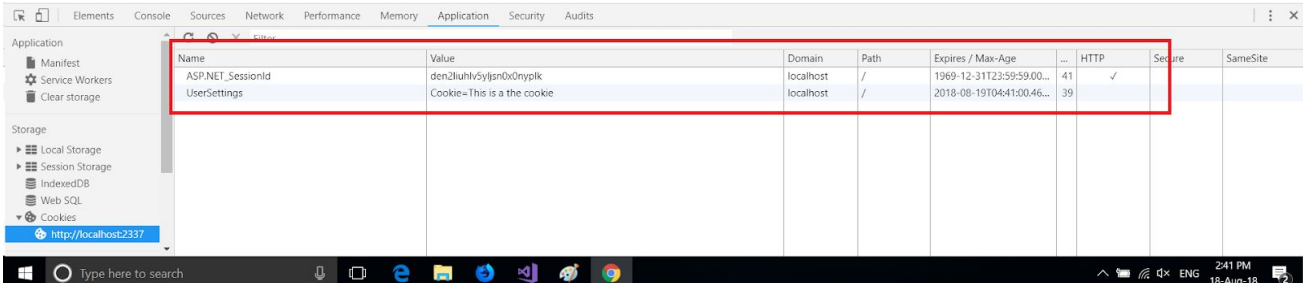


Please remember that there is a difference between Sessions & Cookies.



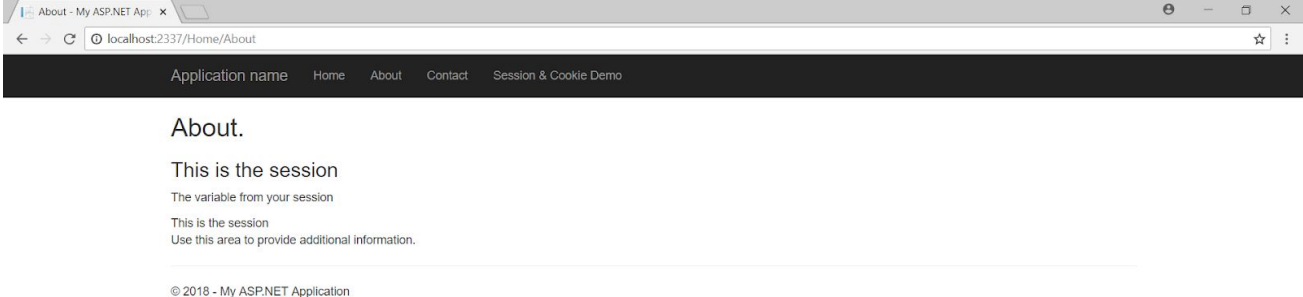
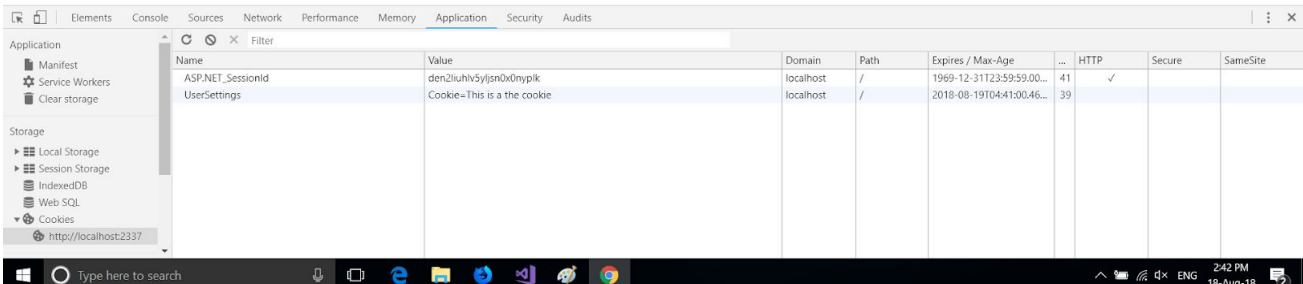
Step 7

You will notice that these values will appear.

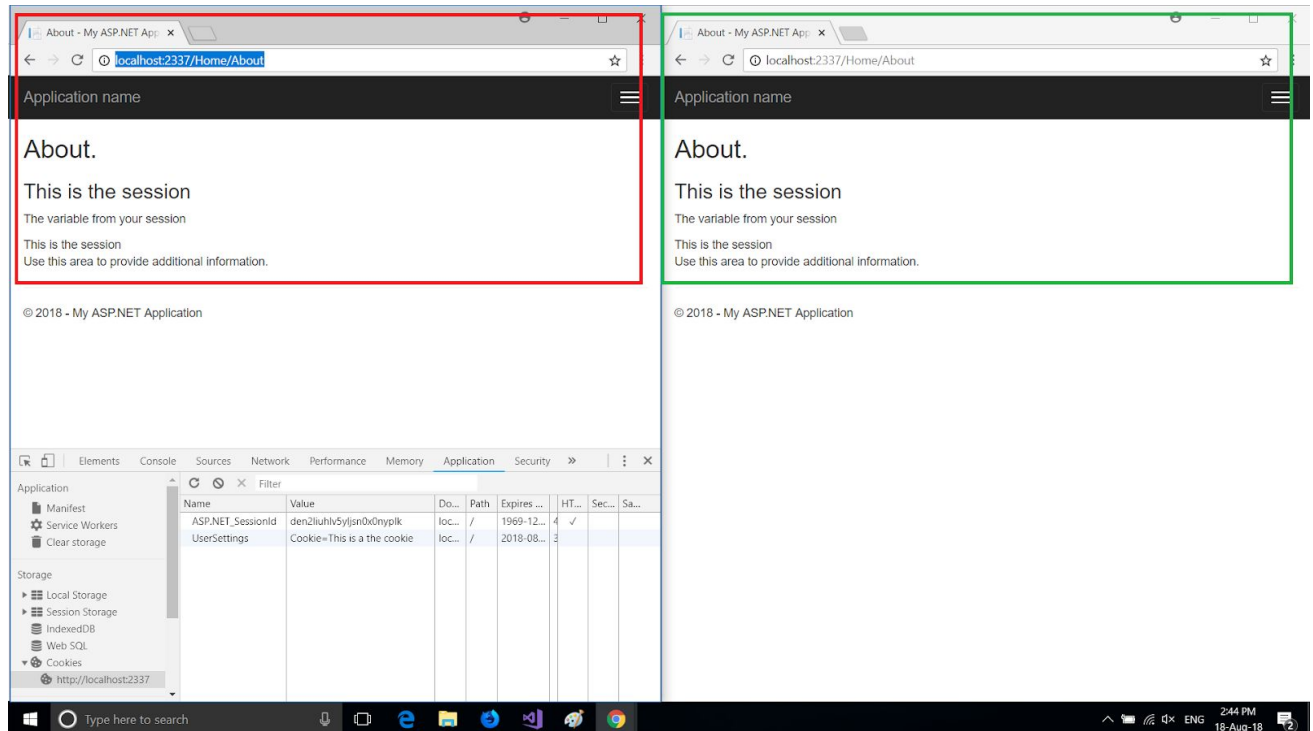
Name	Value	Domain	Path	Expires / Max-Age	HTTP	Secure	SameSite
ASP.NET_SessionId	den2kuhlv5yljsn0x0nyplk	localhost	/	1969-12-31T23:59:59.000...	41	✓	
UserSettings	Cookie=This is a the cookie	localhost	/	2018-08-19T04:41:00.46...	39		

If you navigate to the About page, you will notice the Session value is there too.

Name	Value	Domain	Path	Expires / Max-Age	HTTP	Secure	SameSite
ASP.NET_SessionId	den2kuhlv5yljsn0x0nyplk	localhost	/	1969-12-31T23:59:59.000...	41	✓	
UserSettings	Cookie=This is a the cookie	localhost	/	2018-08-19T04:41:00.46...	39		

If you open another tab in the browser, you will notice that the value of the Session is there as well.



If I go ahead and **close the browser**, and re-run the solution. You would realise that the value from the session is no longer there but the value from the Cookie is still present if you start the application again.

Conclusion

Sessions & Cookies are very important knowledge if you are planning to be a web developer. It is very important to understand the difference between them.