

Build an EF and ASP.NET Core 3.0 App HOL

Lab 10

This lab walks you through updating your site with basic GDPR support. Prior to starting this lab, you must have completed Lab 9.

Part 1: Update Startup.cs

Step 1: Update the Configure method

1) After the call to UseStaticFiles, add UseCookiePolicy:

```
app.UseHttpsRedirection();  
app.UseWebOptimizer();  
app.UseStaticFiles();  
app.UseCookiePolicy();  
app.UseRouting();
```

Step 2: Update the ConfigureServices method

1) After the call to services.AddControllersWithViews(), configure the cookie policy options:

```
services.Configure<CookiePolicyOptions>(options =>  
{  
    // This lambda determines whether user consent for non-essential cookies is  
    // needed for a given request.  
    options.CheckConsentNeeded = context => true;  
    options.MinimumSameSitePolicy = SameSiteMode.None;  
});
```

2) After configuring the cookie policy options, set tempdata and session cookies as essential:

```
// The TempData provider cookie is not essential. Make it essential  
// so TempData is functional when tracking is disabled.  
services.Configure<CookieTempDataProviderOptions>(options => { options.Cookie.IsEssential = true; });  
services.AddSession(options => { options.Cookie.IsEssential = true; });
```

Part 2: Update the Home Controller

Add actions to grant and withdraw consent

Step 1: Update the using statements

- 1) Add the following using statements to the top of the HomeController.cs file:

```
using Microsoft.AspNetCore.Http.Features;
using AutoLot.Web.Extensions;
```

Step 2: Add the GrantConsent action method

- 1) The GrantConsent is used when a user agrees to the cookie policy:

```
[HttpGet]
public IActionResult GrantConsent()
{
    HttpContext.Features.Get<ITrackingConsentFeature>().GrantConsent();
    return RedirectToAction(nameof(Index), nameof(HomeController).RemoveController(),
        new {area = ""});
}
```

Step 3: Add the WithdrawConsent action method

- 1) The WithdrawConsent is used when a user changes their mind about the cookie policy and no longer grants consent:

```
[HttpGet]
public IActionResult WithdrawConsent()
{
    HttpContext.Features.Get<ITrackingConsentFeature>().WithdrawConsent();
    return RedirectToAction(nameof(Index), nameof(HomeController).RemoveController(),
        new {area = ""});
}
```

Part 3: Add/Update the views

Step 1: Add the `_cookieConsentPartial` partial view

- 1) Add a new partial view named `_cookieConsentPartial.cshtml` in the `Views\Shared` folder.
- 2) Update the code to match the following:

```
@using Microsoft.AspNetCore.Http.Features
@{
    var consentFeature = Context.Features.Get<ITrackingConsentFeature>();
    var showBanner = !consentFeature?.CanTrack ?? false;
    var cookieString = consentFeature?.CreateConsentCookie();
}
@if (showBanner)
{
    <div id="cookieConsent" class="alert alert-info alert-dismissible fade show" role="alert">
        Use this space to summarize your privacy and cookie use policy. <a asp-area="" asp-
controller="Home" asp-action="Privacy">Learn More</a>.
        <button type="button" class="accept-policy close" data-dismiss="alert" aria-label="Close"
data-cookie-string="@cookieString">
            <span aria-hidden="true">Accept</span>
        </button>
    </div>
    <script>
        (function() {
            var button = document.querySelector("#cookieConsent button[data-cookie-string]");
            button.addEventListener("click",
                function(event) {
                    document.cookie = button.dataset.cookieString;
                },
                false);
        })();
    </script>
}
```

Step 2: Update the `_Layout` partial view

- 1) In the `_LayoutPartial.cshtml` file (located in `Views\Shared`), update the markup just after the closing `</header>` tag to the following:

```
<div class="container">
    <partial name="_CookieConsentPartial"/>
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>
```

Step 3: Update the _Menu partial view

1) In the _Menu.cshtml file (located in Views\Shared), add the following using statement:

```
@using Microsoft.AspNetCore.Http.Features
```

2) Update the razor block to the following:

```
@{  
    var dealerInfo = DealerInfoMonitor.CurrentValue;  
    var consentFeature = Context.Features.Get<ITrackingConsentFeature>();  
    var showBanner = !consentFeature?.CanTrack ?? false;  
}
```

3) Add the following after the Privacy menu item (before the closing tag):

```
@if (showBanner)  
{  
    <li class="nav-item">  
        <a class="nav-link text-dark" asp-controller="Home" asp-action="GrantConsent"  
            title="Accept Cookie Policy">Accept Cookie Policy <i class="fas fa-cookie-bite"></i></a>  
    </li>  
}  
else  
{  
    <li class="nav-item">  
        <a class="nav-link text-dark" asp-controller="Home" asp-action="WithdrawConsent"  
            title="Revoke Cookie Policy">Revoke Cookie Policy <i class="fas fa-cookie"></i></a>  
    </li>  
}
```

Summary

This lab added basic support for GDPR and completes with workshop.

Next Steps

In the next lab, you will add an MVC Area to the application.