When you are hosting an ASP.NET Core application on IIS and you need to get the username of the user who is accessing the site, you might encounter issues where **Environment.UserName** returns the username of the IIS application pool identity instead of the actual user who is logged in.

To get the username of the user who is accessing the site, you should use the **HttpContext** to retrieve the identity of the user. Here are a few methods to achieve this:

**1. Using HttpContext.User.Identity.Name**

This is the most straightforward way to get the username of the authenticated user.

public class HomeController : Controller

{

private readonly IHttpContextAccessor \_httpContextAccessor;

public HomeController(IHttpContextAccessor httpContextAccessor)

{

\_httpContextAccessor = httpContextAccessor;

}

public IActionResult Index()

{

string username = \_httpContextAccessor.HttpContext.User.Identity.Name;

return View();

}

}

Copied

**2. Using User.Identity in a Razor View**

If you need the username in a Razor view, you can access it directly using **User.Identity.Name**.

@User.Identity.Name

Copied

**3. Using HttpContext.User in a Middleware**

If you need the username in a middleware, you can access it via **HttpContext**.

public class UsernameMiddleware

{

private readonly RequestDelegate \_next;

public UsernameMiddleware(RequestDelegate next)

{

\_next = next;

}

public async Task InvokeAsync(HttpContext context)

{

string username = context.User.Identity.Name;

*// Do something with the username*

await \_next(context);

}

}

Copied

**4. Configuring Windows Authentication**

If you are using Windows Authentication, make sure it is enabled in IIS and in your ASP.NET Core application.

**In IIS:**

1. Select your site in IIS Manager.
2. Double-click "Authentication".
3. Enable "Windows Authentication".
4. Disable "Anonymous Authentication" if you want to force Windows Authentication.

**In Startup.cs:**

Ensure that Windows Authentication is configured in your **Startup.cs** file.

public void ConfigureServices(IServiceCollection services)

{

services.AddAuthentication(IISDefaults.AuthenticationScheme);

services.AddControllersWithViews();

}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

});

}

Copied

**5. Using HttpContextAccessor**

To use **IHttpContextAccessor**, you need to register it in the **Startup.cs** file.

public void ConfigureServices(IServiceCollection services)

{

services.AddHttpContextAccessor();

services.AddControllersWithViews();

}

Copied

**Summary**

By using **HttpContext.User.Identity.Name**, you can get the username of the user who is accessing the site. Ensure that Windows Authentication is properly configured in IIS and your application to get the correct username.