**Section01 Authentication and Authorization**

**Notes: -**

**1-Authentication: means how are you?**

**(in this layer we generate claims and we use authentication configuration to determine the login path and the cookie name generated)**

**2-Authorization: means are you allowed?**

**(In this part it will check for the cookie by analyze the user principle and the user identities and claims)**

**Steps: -**

**1-in the startup.cs we define the authentication layer configuration and we use the authentication and authorization middleware’s as below**

**using Microsoft.AspNetCore.Builder;**

**using Microsoft.AspNetCore.Hosting;**

**using Microsoft.Extensions.DependencyInjection;**

**using Microsoft.Extensions.Hosting;**

**namespace Basics{**

**public class Startup{**

**// This method gets called by the runtime. Use this method to add services to the container.**

**// For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?LinkID=398940**

**public void ConfigureServices(IServiceCollection services){**

**//we have to inject the authenticaiton in order to access the authorization middleware**

**//we see that we specify the login path foreach authorize action in any controller it will redirect to the Authenticate**

**services.AddAuthentication("CookieAuth")**

**.AddCookie("CookieAuth", config =>{**

**config.Cookie.Name = "Grandmas.Cookie";**

**config.LoginPath = "/Home/Authenticate";});**

**//to apply controllers with views**

**services.AddControllersWithViews();}**

**// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.**

**public void Configure(IApplicationBuilder app, IWebHostEnvironment env){**

**if (env.IsDevelopment()){app.UseDeveloperExceptionPage();}**

**//we apply the routing middleware to activate routing to whicle endpoint we want**

**app.UseRouting();**

**//means how are you?**

**app.UseAuthentication();**

**//we have to make sure that the authorization middleware must set after the routing middleware**

**//we have also to inject the authenticaiton cookie to allow pass the autorization middleware**

**//means : are you allowed?**

**app.UseAuthorization();**

**app.UseEndpoints(endpoints =>{endpoints.MapDefaultControllerRoute();});}}}**

**2-on the HomeController.cs we define action that used to redirect for each authorize action which generate user principle from the user identity from the claims**

**using Microsoft.AspNetCore.Authentication;**

**using Microsoft.AspNetCore.Authorization;**

**using Microsoft.AspNetCore.Http;**

**using Microsoft.AspNetCore.Mvc;**

**using System.Collections.Generic;**

**using System.Security.Claims;**

**using System.Threading.Tasks;**

**namespace Basics.Controllers{**

**public class HomeController : Controller{**

**public IActionResult Index(){return View();}**

**//mainly to used guard the action**

**[Authorize]**

**public IActionResult Secret(){return View();}**

**public async Task<IActionResult> Authenticate(){**

**//claim is an implementation not specified to microsoft**

**//claim is the information that put inside the cookie that will be created**

**//we can have multiple claims with multiple identities that can be assigned to the same user principle**

**var grandmaCliams = new List<Claim>(){**

**new Claim(ClaimTypes.Name,"Bob"),**

**new Claim(ClaimTypes.Email,"Bob@gmail.com"),**

**new Claim("Grandma.Says","Very nice boy")};**

**var licenseClaims = new List<Claim>(){**

**new Claim(ClaimTypes.Name,"Bob K Foo"),**

**new Claim("DrivingLicense","A+")};**

**var grandmaIdentity = new ClaimsIdentity(grandmaCliams,"Grandma Identity");**

**var licenseIdentity = new ClaimsIdentity(licenseClaims, "Government");**

**//define the user principle that contains multiple claim identities that each one contains single claim**

**var userPrinciple = new ClaimsPrincipal(new[] { grandmaIdentity, licenseIdentity });**

**//register the user principle and create cookie contains all the user principle that we put it**

**await HttpContext.SignInAsync(userPrinciple);**

**return RedirectToAction("Index");}}}**

**//in the first time when there is no cookie found on the browser and the user browse to the below link** [**localhost:53923/Home/Secret**](http://localhost:53923/Home/Secret)**.**

**//it will redirect to the action authorize which create user principle and store on the cookie.**

**//after this any request to the same link the authorize check if the user is authorizing it will go forward to the Secret action not to the authorize action while the cookie is existing on the browser**