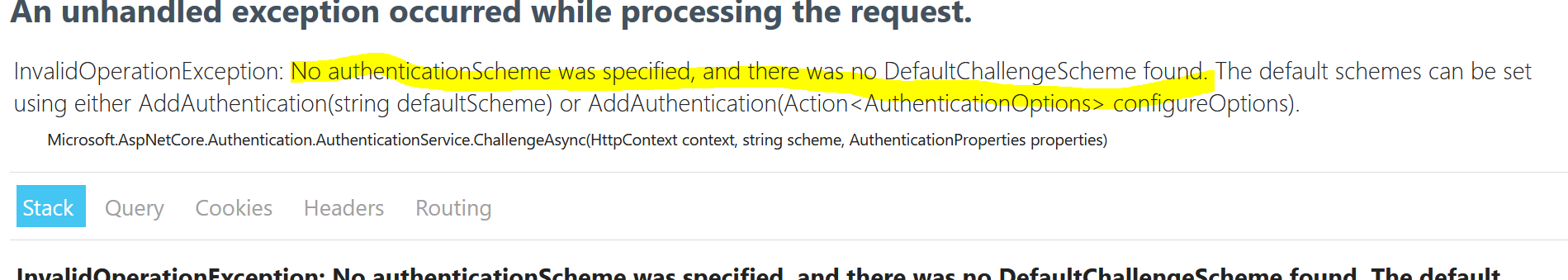
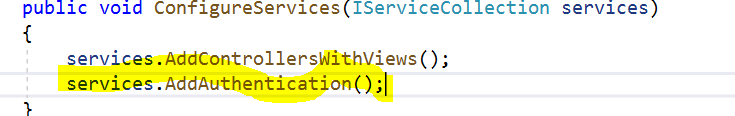
**Authentication & Authorization**

**Auth1**

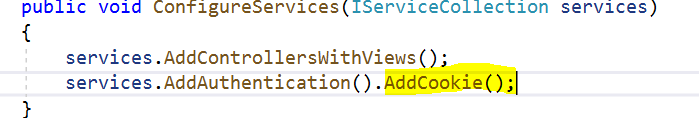
1. **Authentication - > Identity**
2. **Authorization -> Permissions**
3. **Create an application Asp.net core web application named “Auth1”**
4. **Now create a new Link ( Route) named Secured , create a action with name secured and Add** [Authorize]
5. **When we**  run this igt will throw error “ No Authentication Scheme was specified and no DefaultchallengeScheme found”



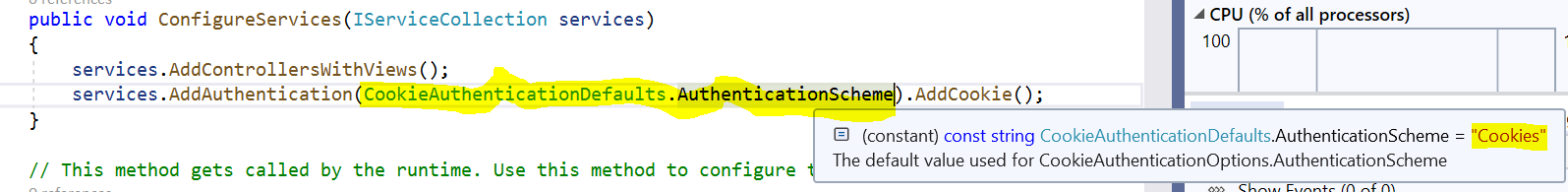
1. Since web is state less , so for each request Authentication is required
2. Every time user will send username and password to server and server then allow to in the system.
3. Or Server will create a ticket on your first request (username/password) and everytime user will show that ticket and server will validate ticket is Authentic or not .
4. In Case of **Cookie Authentication Scheme** , server looking for Cookie.
5. Server prompt for User name and password is Challenge Scheme.
6. Now lets solve #5 issue
7. In Startup.cs



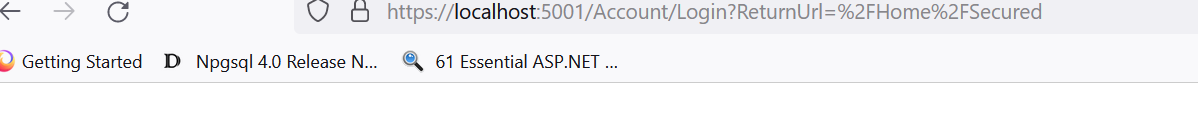
Then will get same error



Then again will get same error.

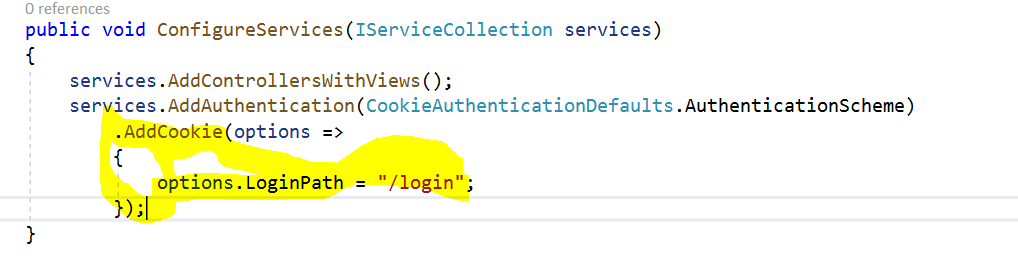


So here we are specifying Authentication scheme name =Cookies , now lets run application

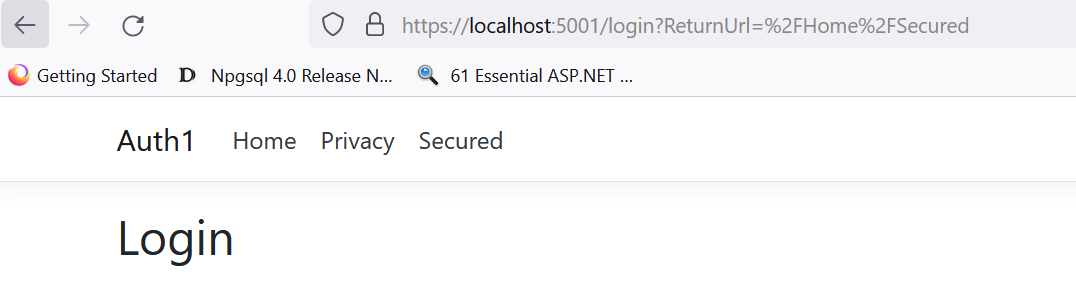


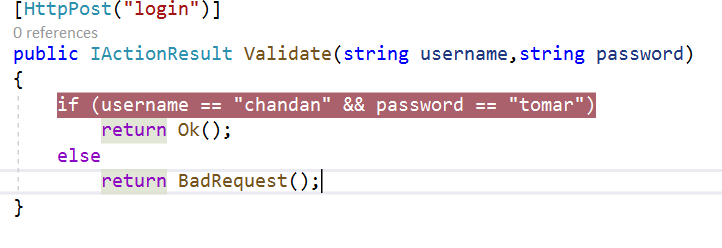
This time will get 404 error page not found , here we can see application looking for Account/Login which doesn’t exists so now lets fix this.

By default it expect Account/login so and add a login action and view

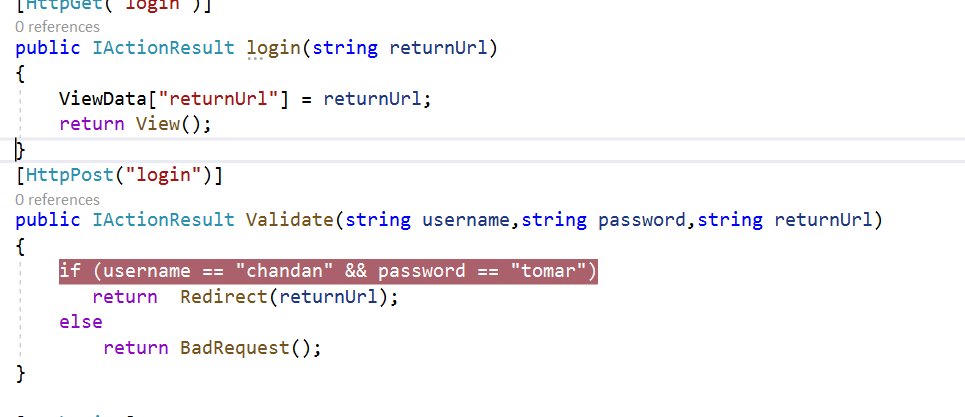


Now when we will click on Secured then due to authorize it check configured services and check cookies based setting and redirect to Login page

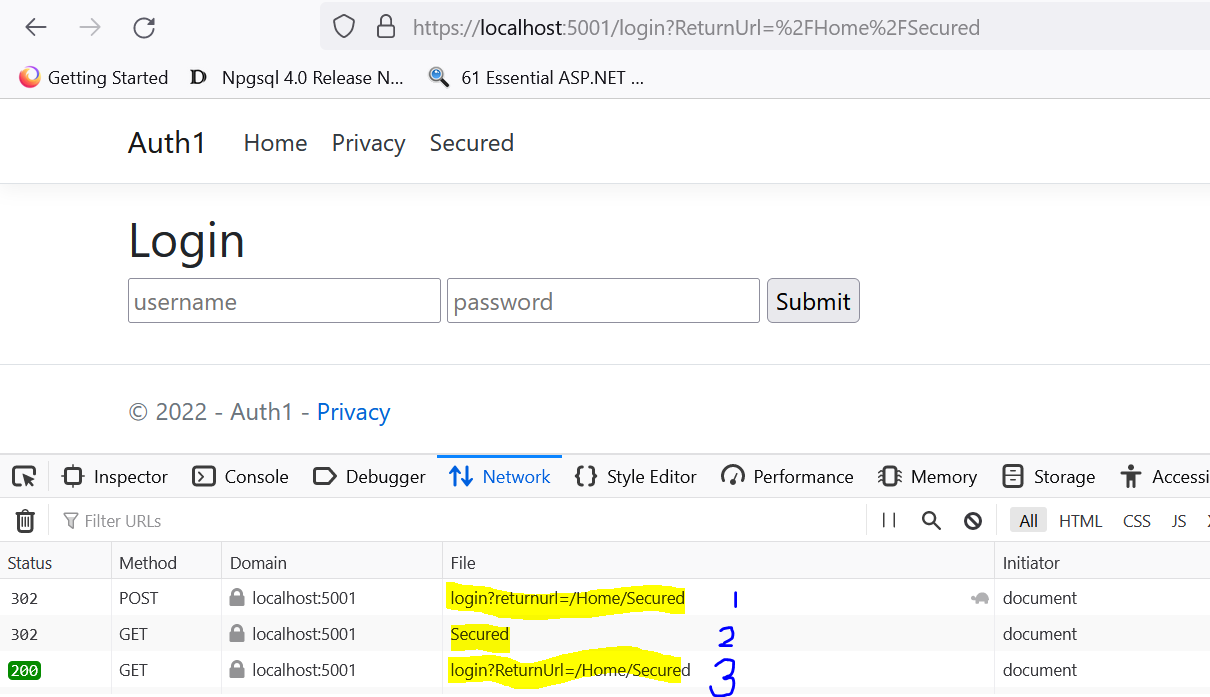




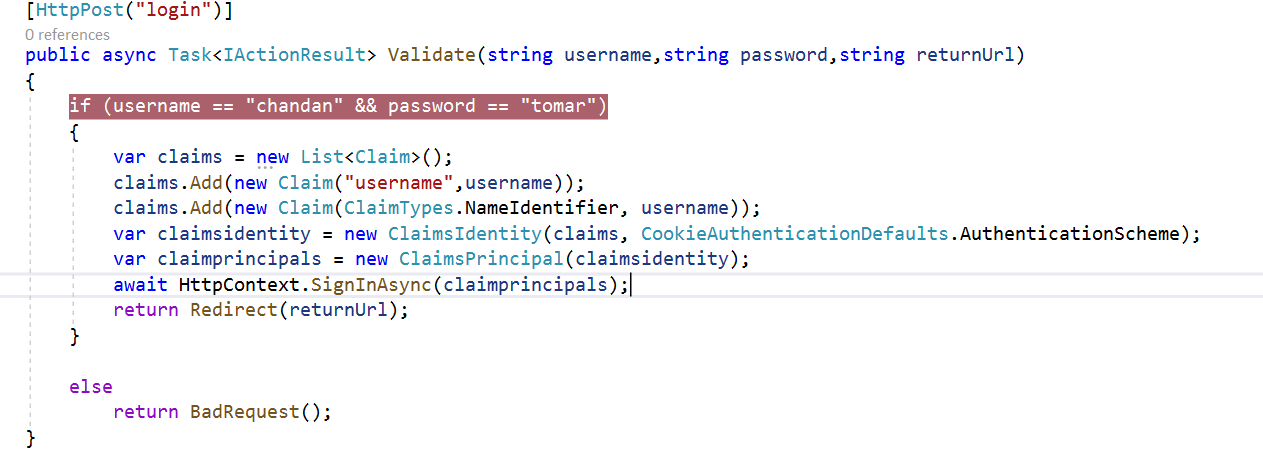
Now after successful login it will be on same page , while After Authorized we got Return Url , so we need to send on that page after successful login.



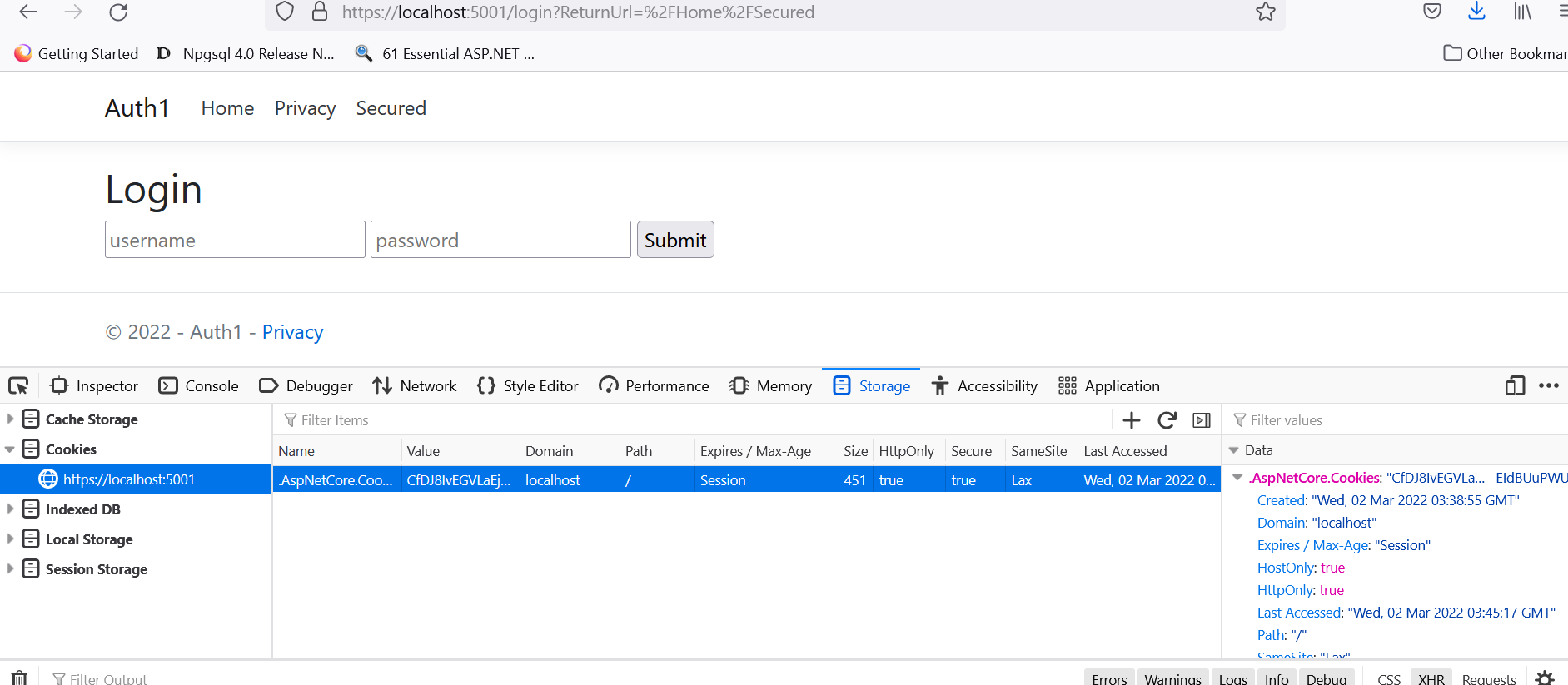
Now check then it again send back to Login page



We can see 1. It redirect to login page , then after save go to secured but again moved to login page .

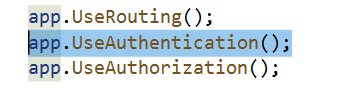


So claim is key value pair of user identity that need to defined after login which create ticket and passed to Context object. If don’t identity any claim then it will redirect to Login page again.

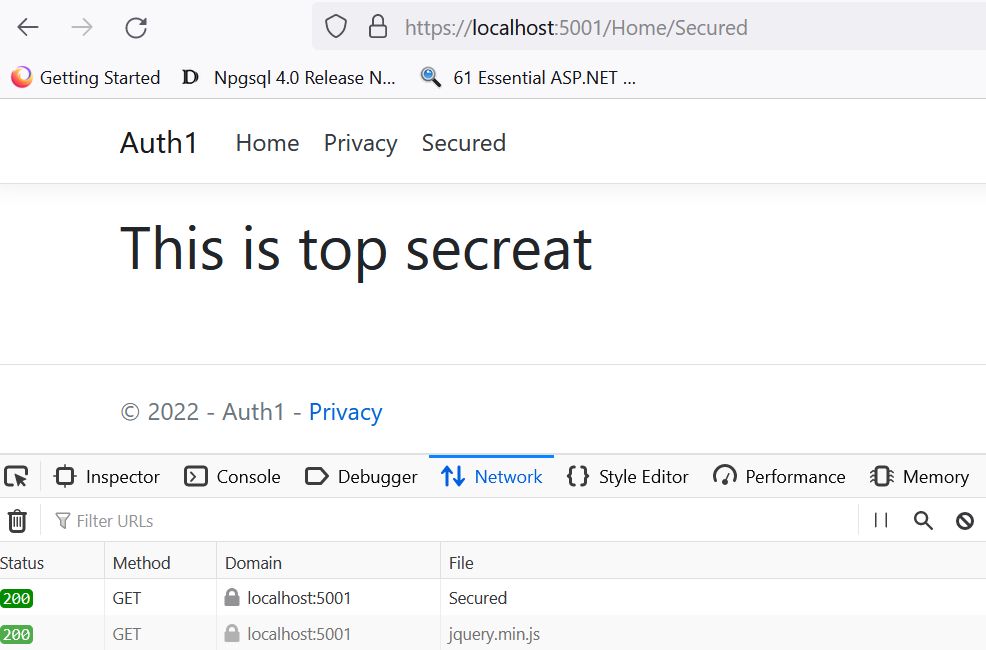


But it will not work because Server has created cookies using identity after login and generate ticket as passed to response , then **Redirect(returnUrl)**  ask to move to secured page but no

Middle ware has been configured so far then again which will fetch from cookies and identify that user is authentic or not



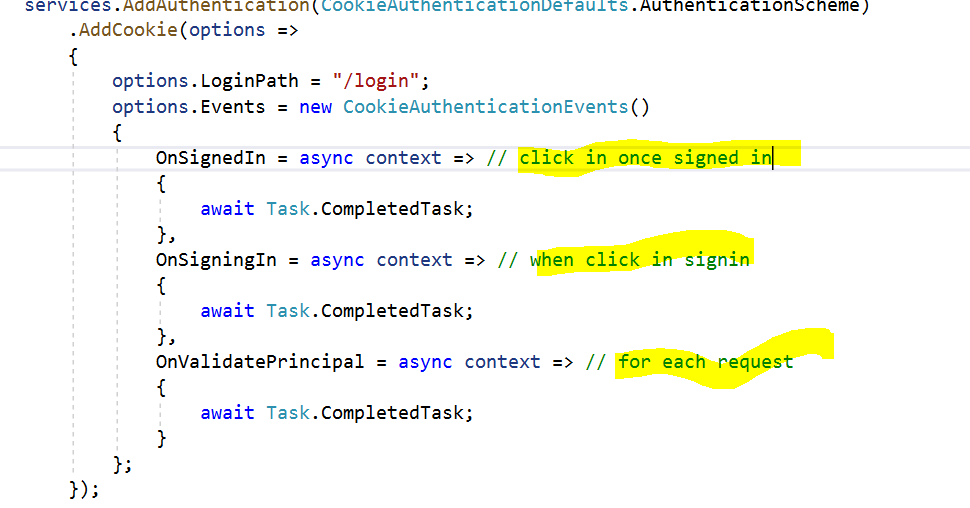
Now lets run again



Create Logout Action there HttpContext.SignoutAsync(); then it will remove cookies from browser.

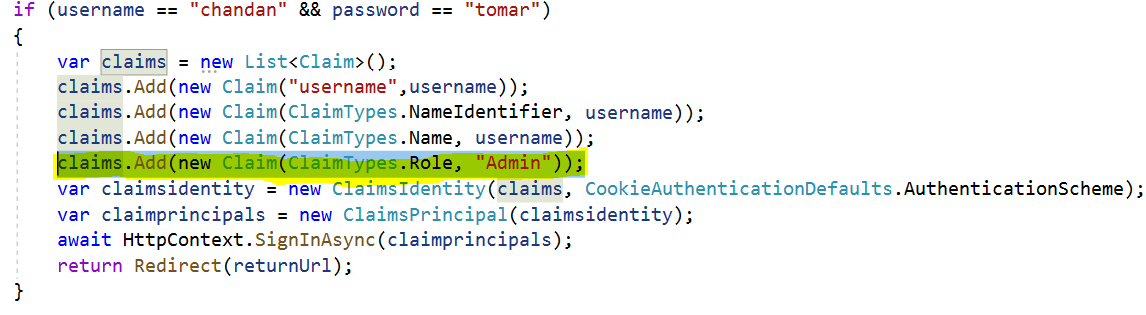
HttpContext.**SignInAsync**(claimprincipals) :  **During this**  process cycle Cookie Authentication Handler do multiple steps. It provide us those event so that we can

Add our execution between them.

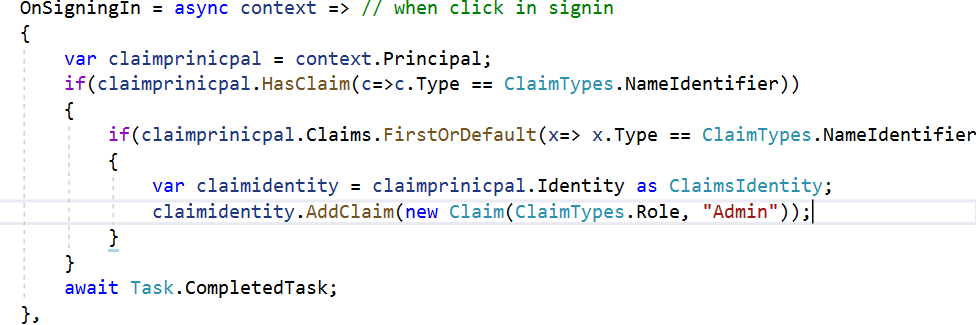


[Authorize] : Once user is Authentic = User.Identity.IsAuthenticated then user will able to access that Action , But suppose we want to give “Admin” role user to access that page .

[Authorize(Roles =”Admin”)] then will give error , so to do that we have to add Roles into Claims.



Or we can add in Startup.cs



**Auth2**

**When**  we think about Challenge Scheme , let first think about below issue when we handle application with below responsibilities

1. One way encryption/hashing password in db
2. Password change/reset options
3. Recovery options
4. Multifactor auth options -SMS or biometric Security

So for identity only we need to give all above in our code , so its better to delegate this task to some other identity server

Like Identity provider are free

Google

Facebook

Twitter

Microsoft

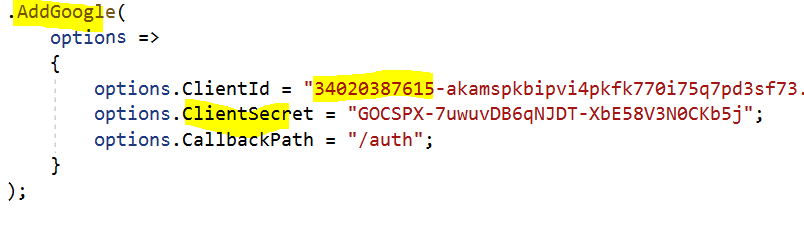
Github

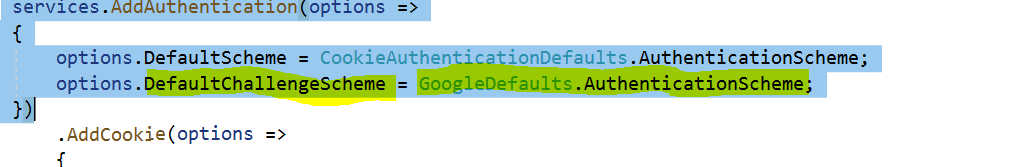
Okta

There are various Auth protocol

1. SAML for old application
2. Auth2.0
3. OpenIDConnect ( its on top of Auth2.0)

Case 1: Google , open console.developer.google.com

1. Create Project in Google , add all details and get Client ID and client secreat
2. Add nuget Package Microsoft.AspnetCore.Authentication.Google
3. Then add following
4. 
5. Then add



This will prompt Google Authentication prompt first by default because we have set Default challenge scheme once logged in it will redirect to page as mentioned in callback



But when we logout it will not logged out google account from browser **because HttpContext.SignOutAsync();** only removed cookies (default) not google .

public IActionResult logout()

{

HttpContext.SignOutAsync();

return Redirect(@"**https://www.google.com/accounts/logout?continue=**

**https://appengine.google.com/\_ah/logout?continue=https://localhost:5001**");

}

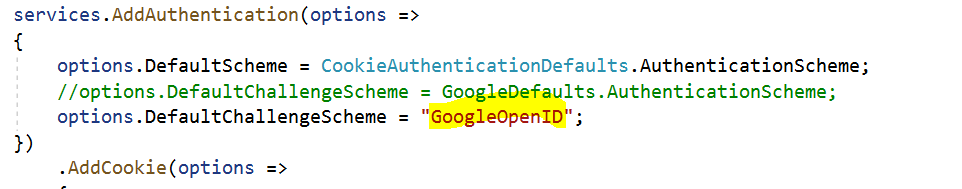
In Above , we are using Google Auth handler , but lets install generic OpenID handler.

Microsoft.Aspnetcore.Authentication.OpenIdConnect



This will work same as like Google Authentication Handler , OpenID Connect handler is build on top of oAuth2.0 while Google Authentication Handler

Works on OAuth2.0 , so OpenID connect provide more claims than Google Handler.



Suppose we want to get token return from OpenID connect then we need to first save into cookies

.AddOpenIdConnect("GoogleOpenID", options =>

{

options.Authority = "https://accounts.google.com";

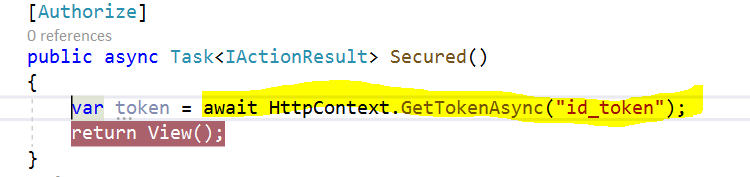
options.ClientId = "34020387615-akamspkbipvi4pkfk770i75q7pd3sf73.apps.googleusercontent.com";

options.ClientSecret = "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";

options.CallbackPath = "/auth";

**options.SaveTokens=true;**

});



This will return JWT token , we can see JWT token details in jwt.io

Now someone modified payload , without modifying signature then it will give “Invalid Signature”