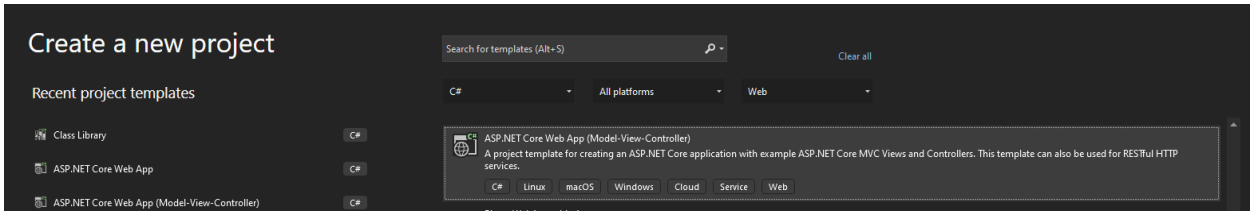
- Tạo một project với template ASP.NET Core Web App – MVC



- Cấu trúc thư mục sẽ có những phần chính như wwwroot (web root folder), Models, Views, Controllers.

- Project bắt đầu chạy từ hàm Main và chạy đến class Startup.

- Trong hàm Configure cấu hình sử dụng các middleware thêm vào pipeline request.
  - app.UseHttpsRedirection() → request tự động chuyển sang https
  - app.UseStaticFiles() → middleware xử lí các request gọi đến file tĩnh, file không thực thi được như html, txt,...
  - app.UseRouting() sẽ định tuyến cho các api request chạy đến các endpoint được map ở bên trong middleware UseEndpoints
  - Ví dụ:

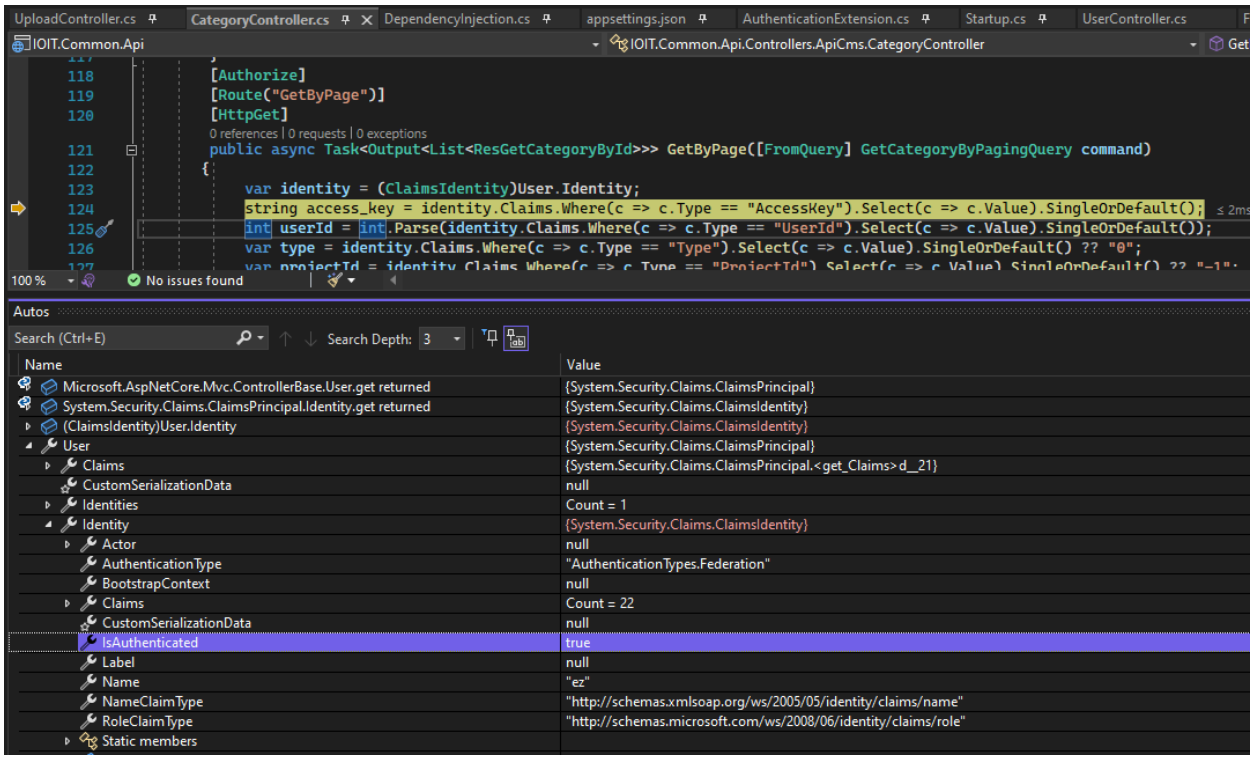
```
15  namespace IOIT.Identity.Api.Controllers.ApiApp
16  {
17      [Route("api/app/[controller]")]
18      [ApiController]
        0 references
19      public class UserController : BaseController
20      {
21          [HttpPost("login")]
            0 references
22          public async Task<IActionResult> Login([FromBody] LoginUserAppQuery command)
23          {
24              var resData = await _mediator.Send(command);
25
26              return Res(new DefaultResponse().Success(
27                  resData,
28                  ApiConstants.MessageResource.ACCTION_SUCCESS,
29                  ApiConstants.StatusCode.Success200
30                  ));
31          }
```

➔ Api login là: /api/app/user/login. Class UserController thì sẽ bỏ Controller đi, đến action login.

- Khi sử dụng middleware UseAuthentication() ta phải đăng ký service và các handle trong hàm ConfigureServices trong class Startup. Ví dụ sử dụng token Bear cho việc xác thực.

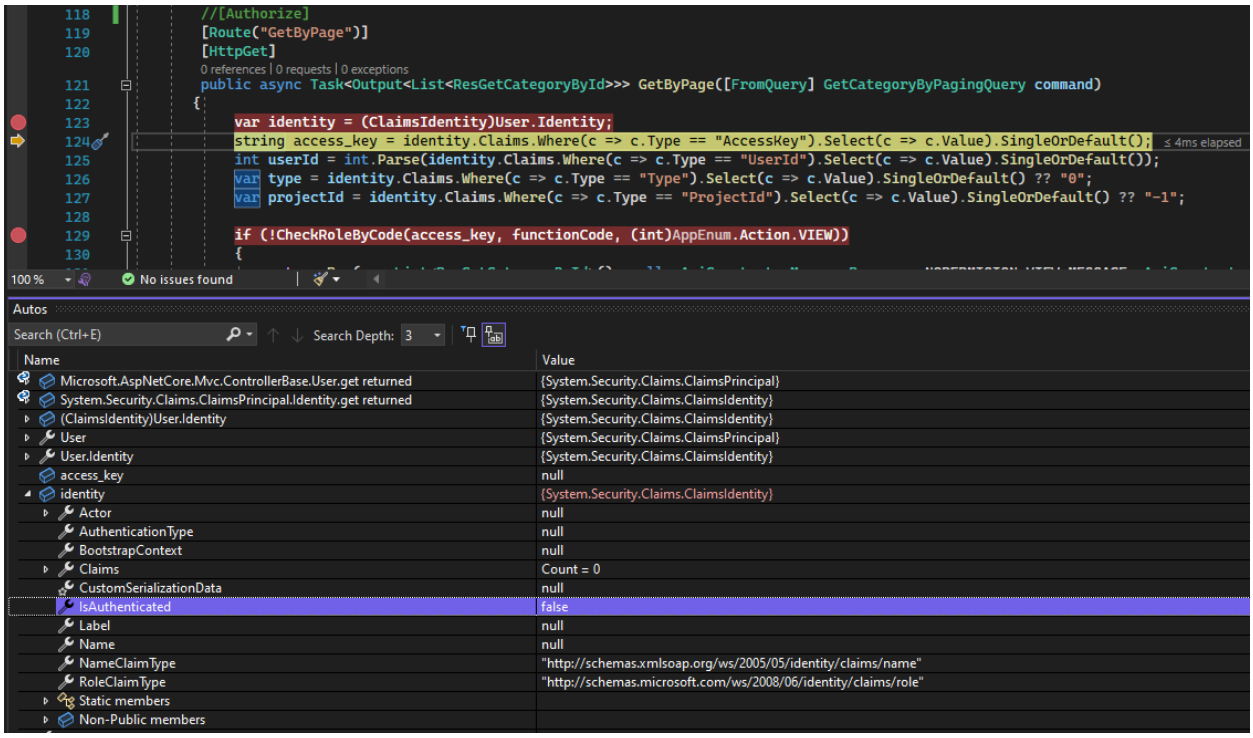
```
249             string domain = Configuration["AppSettings:JwtIssuer"];
250             var authenticationProviderKey = "TestKey";
251             JwtSecurityTokenHandler.DefaultInboundClaimTypeMap.Clear();
252             services.AddAuthentication(options =>
253             {
254                 options.DefaultAuthenticateScheme = authenticationProviderKey;
255                 options.DefaultScheme = JwtBearerDefaults.AuthenticationScheme; // Bear
256                 options.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme; // Bear
257
258             }).AddJwtBearer(authenticationProviderKey, cfg =>
259             {
260                 cfg.RequireHttpsMetadata = false;
261                 cfg.SaveToken = true;
262                 cfg.TokenValidationParameters = new TokenValidationParameters
263                 {
264                     ValidIssuer = domain,
265                     ValidAudience = domain,
266                     IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(Configuration["AppSettings:JwtKey"])),
267                     ClockSkew = TimeSpan.Zero // remove delay of token when expire
268                 };
269             });
```

- Các request sẽ đi qua middleware này, token được validate, thông tin người dùng ở context hiện tại được lưu trong thuộc tính User, đặc biệt quan tâm đến trường isAuthenticated. Nếu token invalid biến isAuthenticated sẽ là false

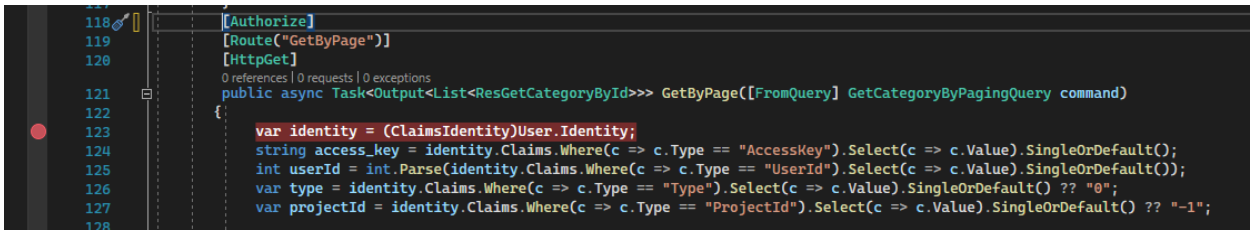


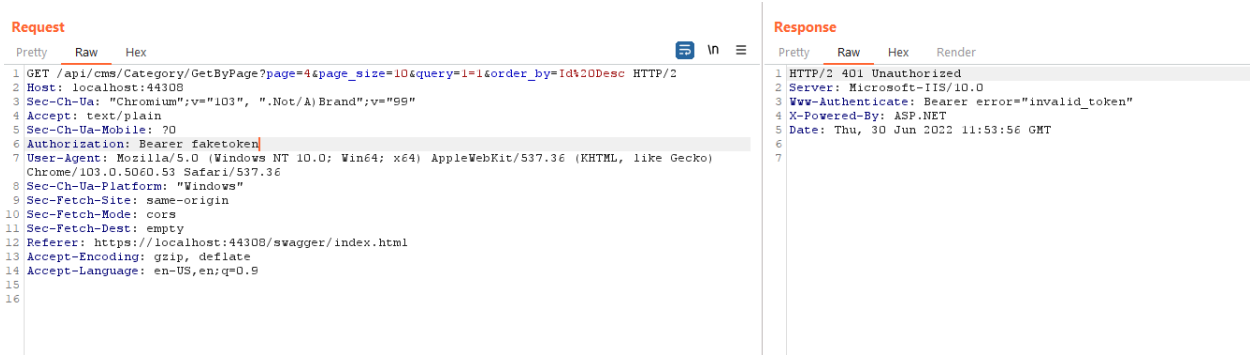
- Khi sử dụng middleware UseAuthorization() và sử dụng thuộc tính [Authorize] trên Controller hoặc trên action, server sẽ kiểm tra biến isAuthenticated trong User identity. Nếu chưa xác thực thì server trả về luôn response UN_AUTHORIZE (Lưu ý: nếu sử dụng thuộc tính [Authorize] nhưng không khai báo middleware UseAuthorization() thì server sẽ không kiểm tra người dùng đã authenticate hay chưa)
- Tùy vào từng request mà có thể đi qua hết các middleware hoặc đi qua 1 vài hoặc ko đi qua middleware nào. Điều kiện để 1 request đi qua 1 middleware là request đó có action gọi đến các handler của middleware. Ví dụ khi sử dụng thuộc tính [Authorize] thì request mới đi qua middleware UseAuthorization; khi claim user identity request sẽ đi qua middleware UseAuthentication.
- Debug demo:

- Khi gửi 1 request với invalid token → người dùng không được xác thực isAuthenticated = false, identity = null



- Khi cho request này đi qua middleware Authorization thì sẽ bị chặn luôn (dòng 118).





- ASP.NET Core Authentication with Cookie: khai báo Middleware và đăng ký Service trong Startup.cs

```
var cookiePolicyOptions = new CookiePolicyOptions
{
    MinimumSameSitePolicy = SameSiteMode.None,
    Secure = CookieSecurePolicy.Always,
    HttpOnly = Microsoft.AspNetCore.CookiePolicy.HttpOnlyPolicy.Always,
};

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();
app.UseAuthentication();
app.UseCookiePolicy(cookiePolicyOptions);
app.UseAuthorization();
```

```
services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)
    .AddCookie(options =>
    {
    options.ExpireTimeSpan = TimeSpan.FromMinutes(20);
    options.SlidingExpiration = true;
    options.AccessDeniedPath = "/Forbidden/";
    });
```

- Trong api login

```
[HttpPost("login")]
0 references
public async Task<ActionResult> Login(string email, string password)
{
    if (ModelState.IsValid)
    {
        var user = 1; //await AuthenticateUser(Input.Email, Input.Password);
        if (user == null)
        {
            ViewBag.erorr = "Login Failed";
            return RedirectToAction("Login");
        }
        var claims = new List<Claim>
        {
            new Claim(ClaimTypes.Name, "nguyenngocdoanh1998@gmail.com"),//user.Email),
            new Claim("FullName", "Nguyen Ngoc Doanh"), //user.FullName),
            new Claim(ClaimTypes.Role, "Administrator"),
        };

        var claimsIdentity = new ClaimsIdentity(
            claims, CookieAuthenticationDefaults.AuthenticationScheme);

        var authProperties = new AuthenticationProperties
        {
            ExpiresUtc = DateTimeOffset.UtcNow.AddMinutes(10),
            IsPersistent = true,
        };

        await HttpContext.SignInAsync(
            CookieAuthenticationDefaults.AuthenticationScheme,
            new ClaimsPrincipal(claimsIdentity),
            authProperties);
```

- Khi gửi request post login → server trả về session

Request

Pretty Raw Hex

```
POST /api/login HTTP/2
Host: localhost:44335
Sec-Ch-Ua: "Chromium";v="103", ".Not/A)Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/103.0.5060.53 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Content-Length: 0
```

Response

Pretty Raw Hex Render

```
HTTP/2 200 OK
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Type: text/html; charset=utf-8
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Vary: Accept-Encoding
Server: Microsoft-IIS/10.0
Set-Cookie: .AspNetCore.Cookies=
CfDJ80WUDEy7yHtAoCG3WAlVwrmzeXU4cfiLpDMeKQD_l5VDRjbZrHSLZrFQN32K7fPz7GyGI
exnJMY-vklkDWTovKYYmr0u5thVsDb800y2GzKr-_xT7gZclWqrMS-YlfUBFoe2keMxI1fcNk
CRyxpbgKp_juBk_6GyqwONggLjGHqYDs0_5zL3s3lFmVPNr4hUkh5SDtu0u0RBgtvDE1GLii7
6RZgz_vlHrLZLxq69bHQ_TnnHyranBcheltUNSULUjmEpoqZVS8zQteC-W8YxAuB8RkfTxiPR
v0GeZiDXUhpEC3LU3gMG6016jurjVFWX1E0QwBJKEheE06nIfUo1Qc4FJhBIyD-4xVilsBIok
tfwBuXgwrLhVaNZhv-WdiLZY-oPk00auulBhgtwvRwHdj7opmIY_WGuTcN1f-fLqYcpFmzLAV
-AUxRag-Tj-c0rs6rQ05eIqmJayZcBUQnuUS-pew3M80rzkEjcUoYDjLoSB8nuR1G_u2mTu-i
KSxYXzQ; expires=Sat, 09 Jul 2022 09:08:21 GMT; path=/; secure;
samesite=lax; httponly
Set-Cookie: .AspNetCore.Antiforgery.G-pwIooZ2YU=
CfDJ80WUDEy7yHtAoCG3WAlVwrmLys188oPiPgCtxH9980RkjEyRRyQ6CNaD4isJFGmCbH1Fy
6nBH7xiXisEY6ECEXh90A6PTTrlvGBywnszjqKR9-3_XI80soguheAC_KG28Q9tiW9s-ZKuk3
3Z-U2R5oo; path=/; secure; samesite=strict; httponly
X-Frame-Options: SAMEORIGIN
X-Powered-By: ASP.NET
Date: Sat, 09 Jul 2022 08:58:21 GMT
```

- Sử dụng Cookie này cho việc xác thực người dùng trong các request sau

Request

Pretty | Raw | Hex

```
1  GET /api/register HTTP/2
2  Host: localhost:44335
3  Sec-Ch-Ua: "Chromium";v="103", ".Not/A)Brand";v="99"
4  Sec-Ch-Ua-Mobile: ?0
5  Sec-Ch-Ua-Platform: "Windows"
6  Upgrade-Insecure-Requests: 1
7  Cookie: .AspNetCore.Cookies=
   CfDJ80WUDEy7yHtAo2G3WAlVwrmzeXU4cfiLpDMeKQD_l5VDRjbZrHSLZrFQN32K7fPz7GyG
   IexnJMY-vklkDWTovKYYmr0u5thVsDb80Oy2GzKr-_xT7gZclWqrM9-YlfUBFoe2keMxIlfc
   NkCRyxpbgkp_juBk_6GyqwONggLjGHqYDs0_5zL3s3lFmVPNr4hUkh5SDtu0u0RBgtvDE1GL
   ii76RZgz_vlHrLZLxq69bHQ_TnnHyranBcheltUNSULUjmEpoqZVS8zQteC-W8YxAuB8RkfT
   xiPRv0GeZiDXUhpEC3LU3gMG60l6jurjVFWX1EOQwBJKEheE06nIfUolQc4FJhBIyD-4xVil
   sBIoktfwBuXgwrLhVaNZhv-WdiLZY-oPk00auulBhgtwvRwHdj7opmIY_WGuTcNlf-fLqYcp
   FmzLAV-AUxRag-Tj-c0rs6rQ05eIqmJayZcBUQnuUS-pew3M80rzkEjcUoYDjLoSB8nuR1G_
   u2mTu-iK9xYXzQ
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/103.0.5060.53 Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/w
   ebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16
17
```

- Kết quả: người dùng được xác thực, claim được thông tin User đã login

User	{System.Security.Claims.ClaimsPrincipal}
Claims	{System.Security.Claims.ClaimsPrincipal.<get_Claims>d__22}
CustomSerializationData	null
Identities	Count = 1
Identity	{System.Security.Claims.ClaimsIdentity}
Actor	null
AuthenticationType	"Cookies"
BootstrapContext	null
Claims	{System.Security.Claims.ClaimsIdentity.<CombinedClaimsIterator>d__38}
CustomSerializationData	null
ExternalClaims	Count = 0
IsAuthenticated	true
Label	null
Name	"nguyenngocdoanh1998@gmail.com"
NameClaimType	"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
RoleClaimType	"http://schemas.microsoft.com/ws/2008/06/identity/claims/role"

Claims	{System.Security.Claims.ClaimsIdentity.<CombinedClaimsIterator>d__38}
System.Collections.Gen...	null
System.Collections.IEnu...	null
Results View	Expanding the Results View will enumerate the IEnumerable
[0]	{http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name: nguyenngocdoanh1998@gmail.com}
[1]	{FullName: Nguyen Ngoc Doanh}
[2]	{http://schemas.microsoft.com/ws/2008/06/identity/claims/role: Administrator}

- Reference:
  - https://docs.microsoft.com/en-us/aspnet/core/security/authentication/cookie?view=aspnetcore-6.0
  - https://github.com/doanhnn/MVC.Api.git