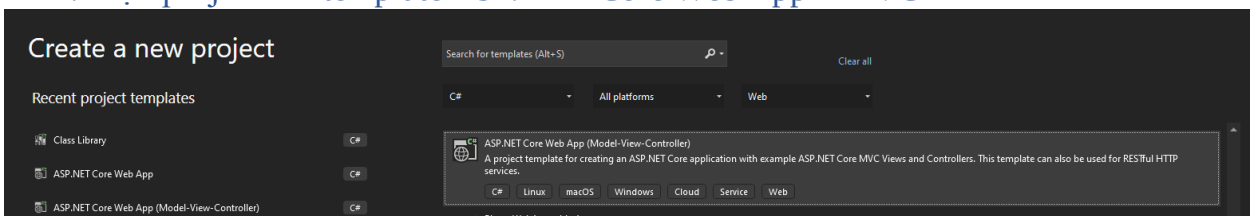
Contents

I. Authentication và Authorization cơ bản trong ASP.NET core MVC

1. Tạo 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.

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
        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?}");
    });
}
}
```

- Trong hàm Configure cấu hình sử dụng các middleware thêm vào pipeline request.
  - app.UseHttpsRedirection() à request tự động chuyển sang https
  - app.UseStaticFiles() à 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:

```
namespace IOIT.Identity.Api.Controllers.ApiApp
{
    [Route("api/app/[controller]")]
    [ApiController]
    0 references
    public class UserController : BaseController
    {
        [HttpPost("login")]
        0 references
        public async Task<IActionResult> Login([FromBody] LoginUserAppQuery command)
        {
            var resData = await _mediator.Send(command);

            return Res(new DefaultResponse().Success(
                resData,
                ApiConstants.MessageResource.ACCTION_SUCCESS,
                ApiConstants.StatusCode.Success200
                ));
        }
```

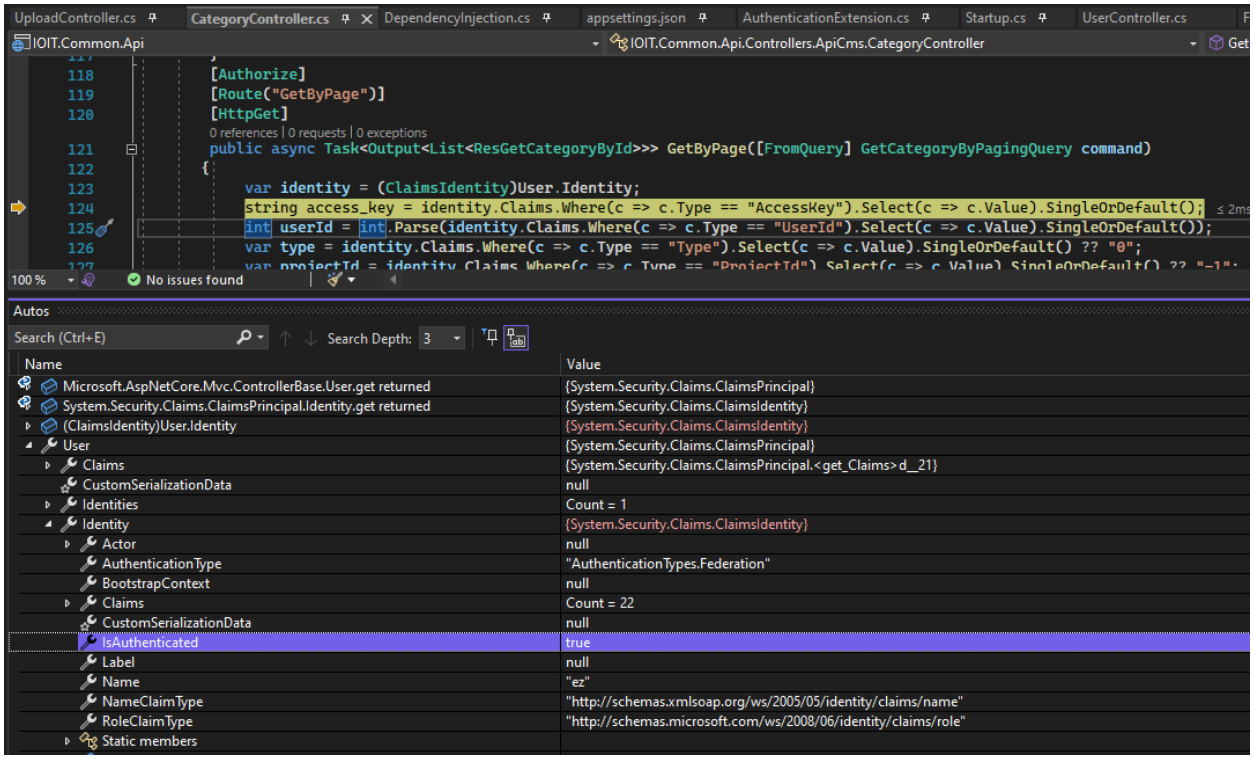
➔ Api login là: /api/app/user/login. Class UserController thì sẽ bỏ Controller đi, đến action login.

2. App.useAuthentication() – with Bearer token

- 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



3. App.useAuthorization()

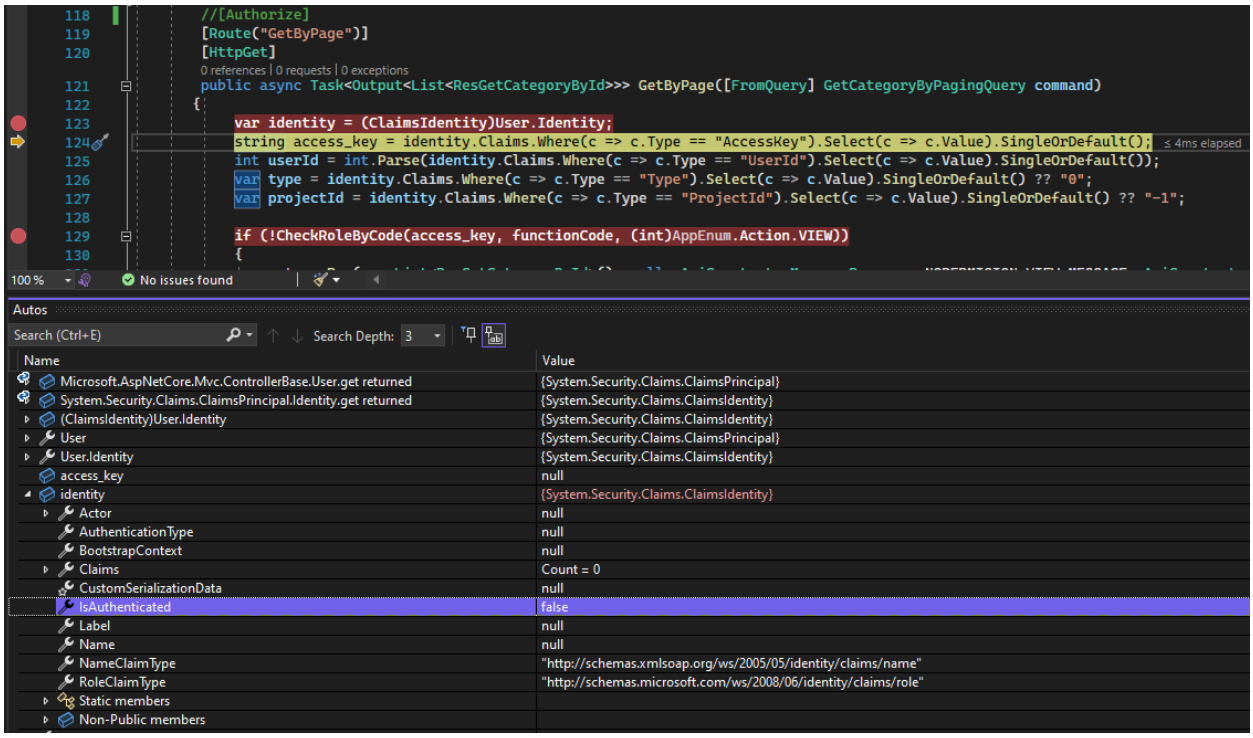
- Khi sử dụng middleware UseAuthorization() và sử dụng thuộc tính [Authorize] trên Controller hoặc trên action, server sẽ kiểm tra biến isAuthenticated trong User identity. Nếu chưa xác thực thì server trả về luôn response UN_AUTHORIZE (Lưu ý: nếu sử dụng thuộc tính [Authorize] nhưng không khai báo middleware 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.
    - Request sẽ đi qua các middleware cho đến khi có Reponse tương ứng được trả về.

- Các Request có thể đi qua các middleware mà ko có sự tác động gì đến HttpContext vì request ko gọi đến Handler nào trong middleware đó.
- 1 request bị tác động khi đi qua 1 middleware khi request đó có action gọi đến các handler của middleware.
- Ví dụ khi sử dụng thuộc tính [Authorize] thì request đi qua middleware UseAuthorization và gọi đến AuthorizationHandler; khi claim user identity request sẽ đi qua middleware UseAuthentication và gọi đến AuthenticationHandler (verify token, cookies, claim user identity).
- Các Request thường bị rẽ nhánh để đến middleware cuối (trả về Response) khi đi qua middleware useEndpoints() (hoặc các middleware có chức năng map request đến action trong api controller)

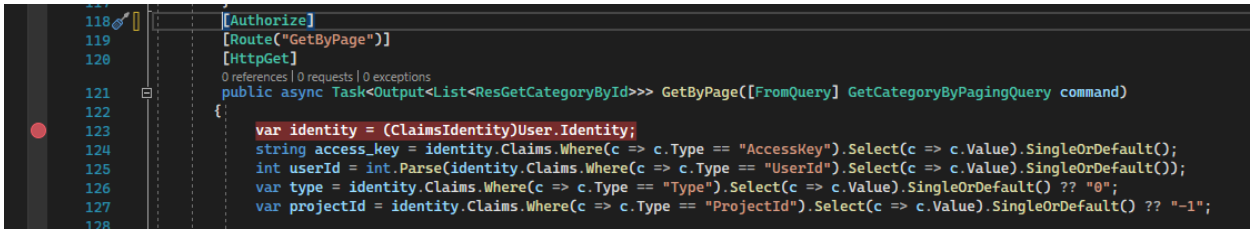
- Đọc thêm middleware trong ASP.NET core: https://xuanthulab.net/asp-net-core-tao-middleware-va-dang-ky-vao-pipeline-cua-ung-dung-web-c-csharp.html

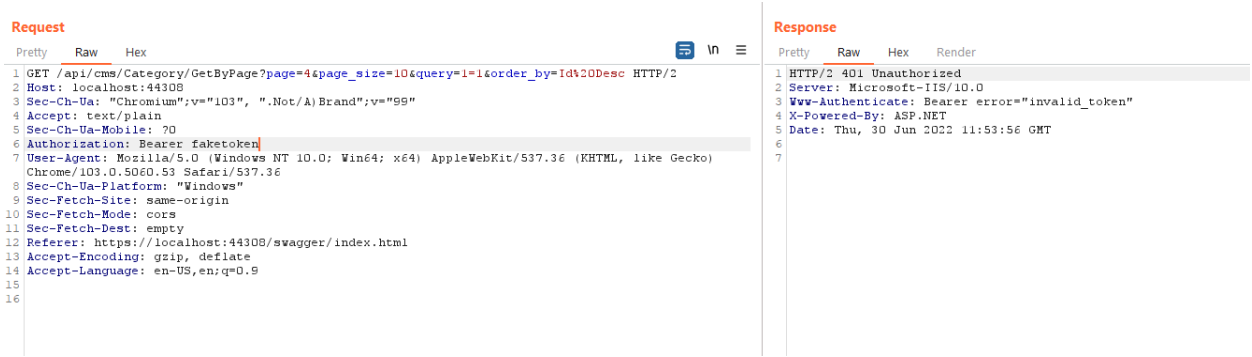
4. Debug demo:

- Khi gửi 1 request với invalid token à 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).





II. ASP.NET Core Authentication with Cookie:

- Khai báo Middleware và đăng ký Service trong Startup.cs

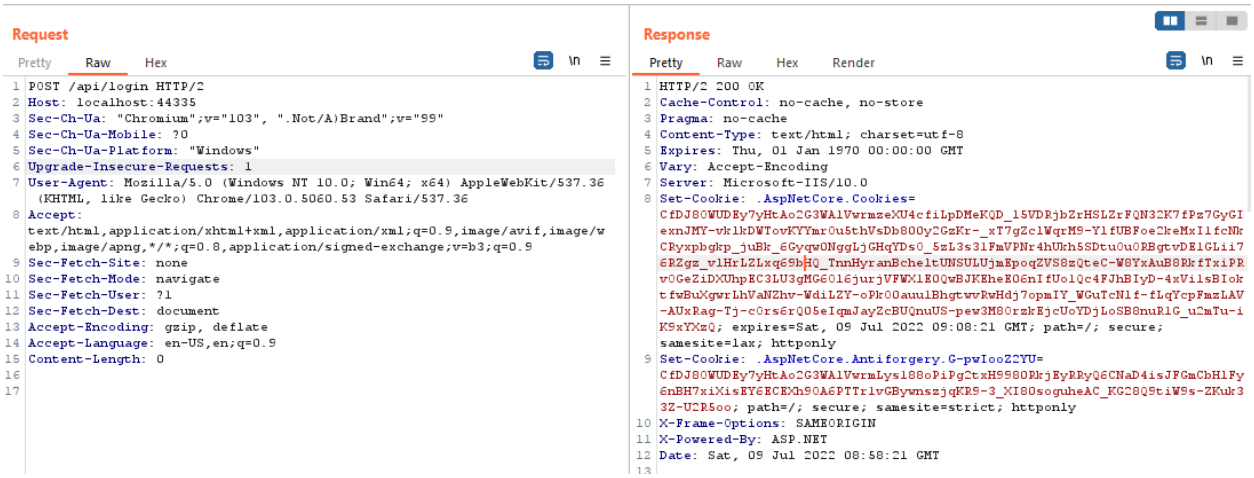
```
var cookiePolicyOptions = new CookiePolicyOptions
{
    MinimumSameSitePolicy = SameSiteMode.None,
    Secure = CookieSecurePolicy.Always,
    HttpOnly = Microsoft.AspNetCore.CookiePolicy.HttpOnlyPolicy.Always,
};

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 à server trả về cookies



Request

Pretty Raw Hex

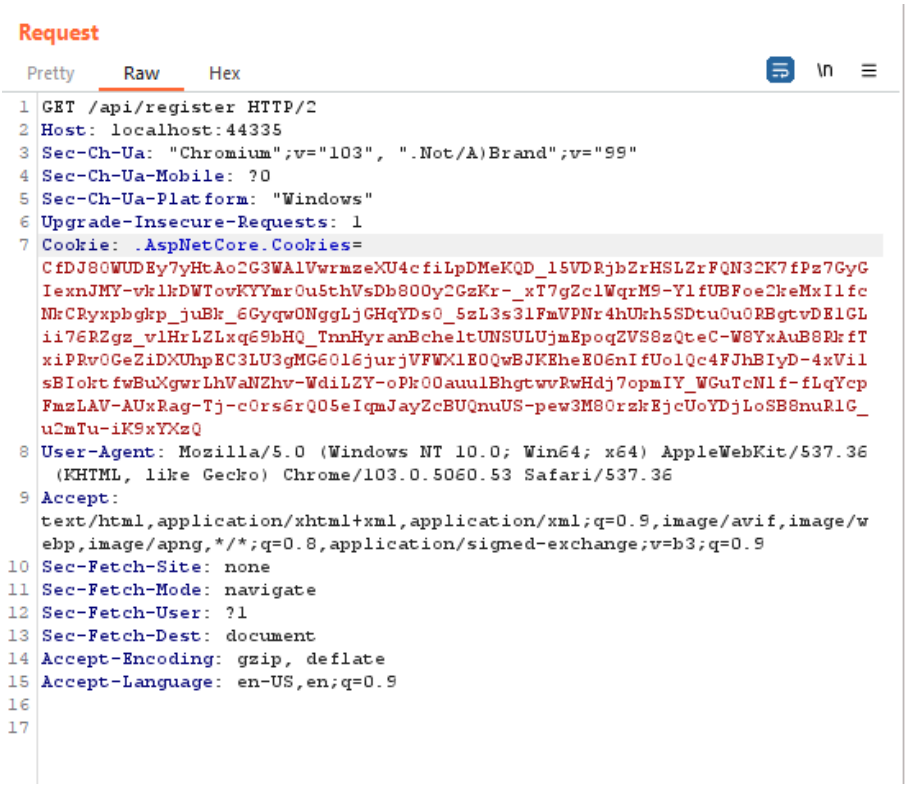
```
1 POST /api/login HTTP/2
2 Host: localhost:44335
3 Sec-Ch-Ua: "Chromium";v="103", ".Not/A)Brand";v="99"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/103.0.5060.53 Safari/537.36
8 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
9 Sec-Fetch-Site: none
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Accept-Encoding: gzip, deflate
14 Accept-Language: en-US,en;q=0.9
15 Content-Length: 0
16
17
```

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Cache-Control: no-cache, no-store
3 Pragma: no-cache
4 Content-Type: text/html; charset=utf-8
5 Expires: Thu, 01 Jan 1970 00:00:00 GMT
6 Vary: Accept-Encoding
7 Server: Microsoft-IIS/10.0
8 Set-Cookie: .AspNetCore.Cookies=
  CfDJ80WUDEy7yHtAo2G3WAlVwrmzeXU4cfiLpDMeKQD_l5VDRjbZrHSLZrFQN32K7fPz7GyGI
  exnJMY-vklkDWTovKYYmr0u5thVsDb800y2GzKr-_xT7gZclWqrM9-YlfUBFoe2keMxI1fcNk
  CRyxpbgkp_juBk_6GyqwONggLjGHqYDs0_5zL3s3lFmVPNr4hUkh5SDtu0u0RBgtvDE1GLii7
  6RZgz_vlHrLZLxq69bHQ_TnnHyranBcheltUNSULUjmEpoqZVS8zQteC-W8YxAuB8RkfTxiPR
  v0GeZiDXUhpEC3LU3gMG60l6jurjVFWX1E0QwBJKEheE06nIfUo1Qc4FJhBIyD-4xVilsBIok
  tfwBuXgwrLhVaNZhv-WdiLZY-oPk00auulBhgtwvRwHdj7opmIY_WGuTcN1f-fLqYcpFmzLAV
  -AUxRag-Tj-c0rs6rQ05eIqmJayZcBUQnuUS-pew3M80rzkEjcUoYDjLoSB8nuR1G_u2mTu-i
  K9xYXzQ; expires=Sat, 09 Jul 2022 09:08:21 GMT; path=/; secure;
  samesite=lax; httponly
9 Set-Cookie: .AspNetCore.Antiforgery.G-pwIooZ2YU=
  CfDJ80WUDEy7yHtAo2G3WAlVwrmLys188oPiPgCtxH9980RkjEyRRyQ6CNaD4isJFGmCbHlFy
  6nBH7xiXisEY6ECEXh90A6PTTrlvGBywnszjqKR9-3_XI80soguheAC_KG28Q9tiW9s-ZKuk3
  3Z-U2R5oo; path=/; secure; samesite=strict; httponly
10 X-Frame-Options: SAMEORIGIN
11 X-Powered-By: ASP.NET
12 Date: Sat, 09 Jul 2022 08:58:21 GMT
13
```

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

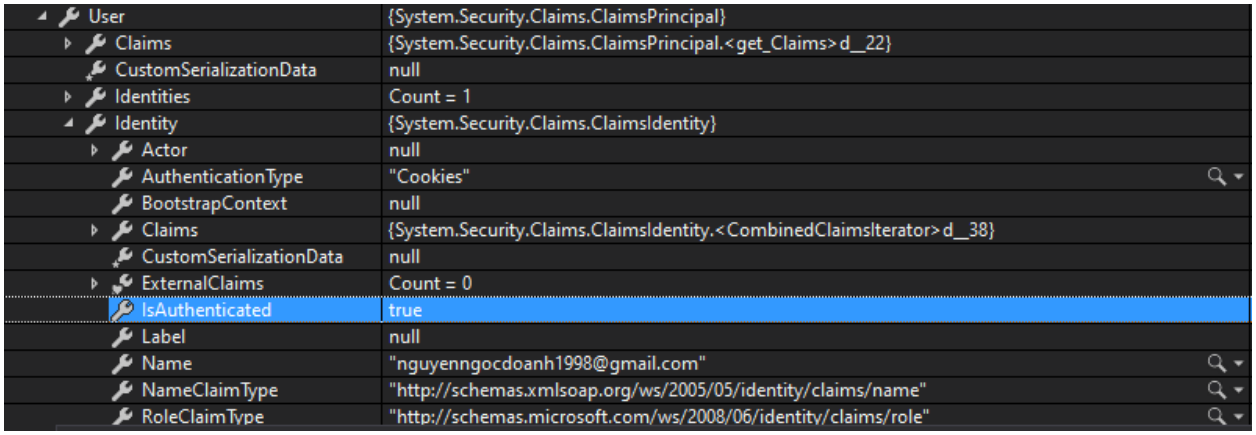


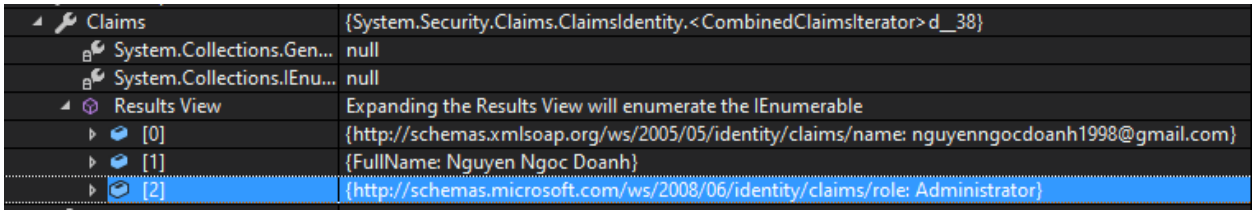
Request

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-vklkDWTovKYYmr0u5thVsDb800y2GzKr-_xT7gZclWqrM9-YlfUBFoe2keMxI1fc
  NkCRyxpbgkp_juBk_6GyqwONggLjGHqYDs0_5zL3s3lFmVPNr4hUkh5SDtu0u0RBgtvDE1GL
  ii76RZgz_vlHrLZLxq69bHQ_TnnHyranBcheltUNSULUjmEpoqZVS8zQteC-W8YxAuB8RkfT
  xiPRv0GeZiDXUhpEC3LU3gMG60l6jurjVFWX1E0QwBJKEheE06nIfUo1Qc4FJhBIyD-4xVil
  sBIoktfwBuXgwrLhVaNZhv-WdiLZY-oPk00auulBhgtwvRwHdj7opmIY_WGuTcN1f-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





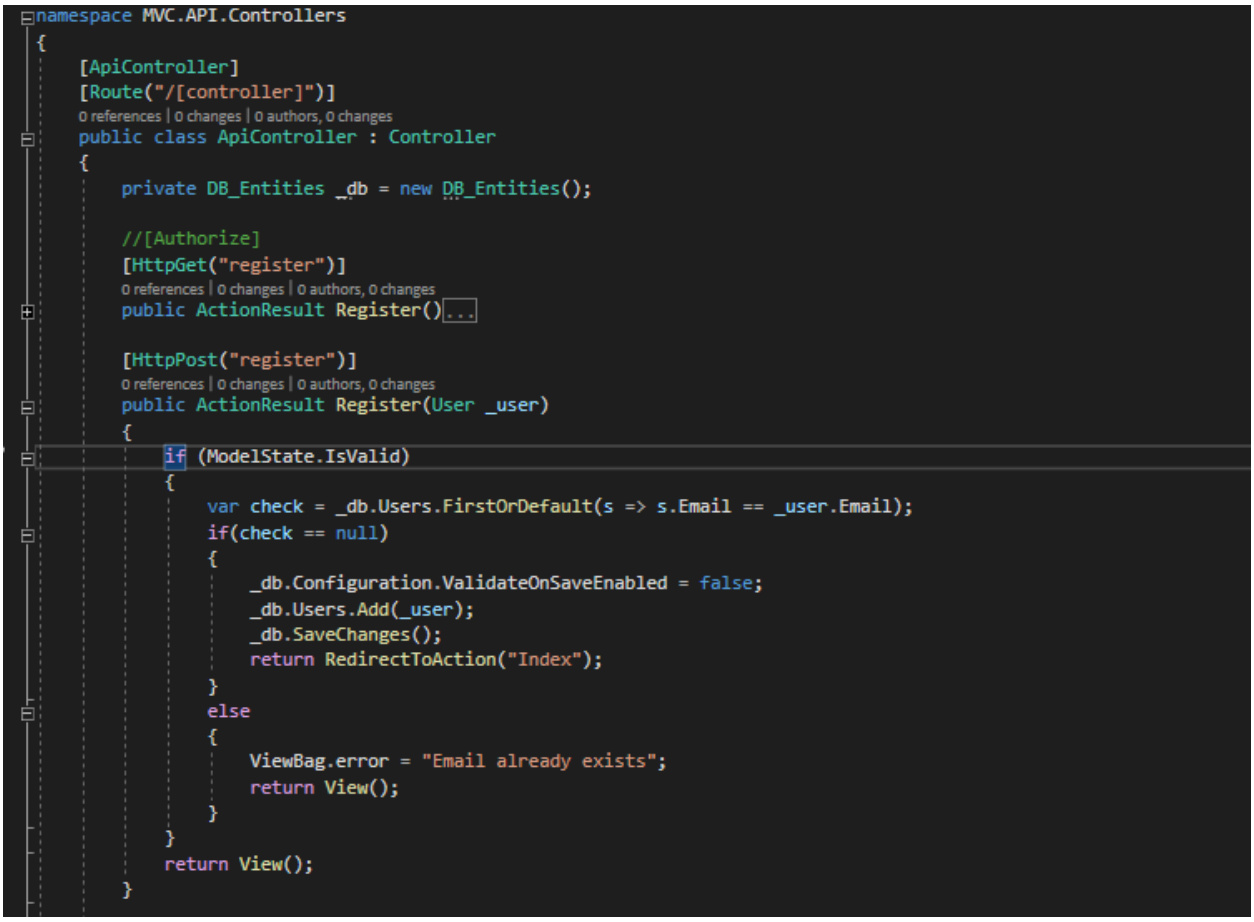
- Reference:
  a. https://docs.microsoft.com/en-us/aspnet/core/security/authentication/cookie?view=aspnetcore-6.0
  b. https://github.com/doanhnn/MVC.Api.git

III. Một số vấn đề khác khi review source code

1. Entity Framework (EF)

```
namespace MVC.API.Entities
{
    3 references | 0 changes | 0 authors, 0 changes
    public class DB_Entities : DbContext
    {
        1 reference | 0 changes | 0 authors, 0 changes
        public DB_Entities() : base("Server=localhost\\SQLEXPRESS;Database=master;Trusted_Connection=True;") { }

        2 references | 0 changes | 0 authors, 0 changes
        public DbSet<User> Users { get; set; }

        0 references | 0 changes | 0 authors, 0 changes
        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            modelBuilder.Entity<User>().ToTable("Users");
            modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
            base.OnModelCreating(modelBuilder);
        }
    }
}
```

- using System.Data.Entity;
- using System.Data.Entity.ModelConfiguration.Conventions;
- Hàm khởi tạo sẽ kết nối đến SQLServer, database master sử dụng connectionString "`Server=localhost\\SQLEXPRESS;Database=master;Trusted_Connection=True;`"
- DbSet<User> Users → bảng Users có model User
- Tạo 1 đối tượng DB_Entities và sử dụng



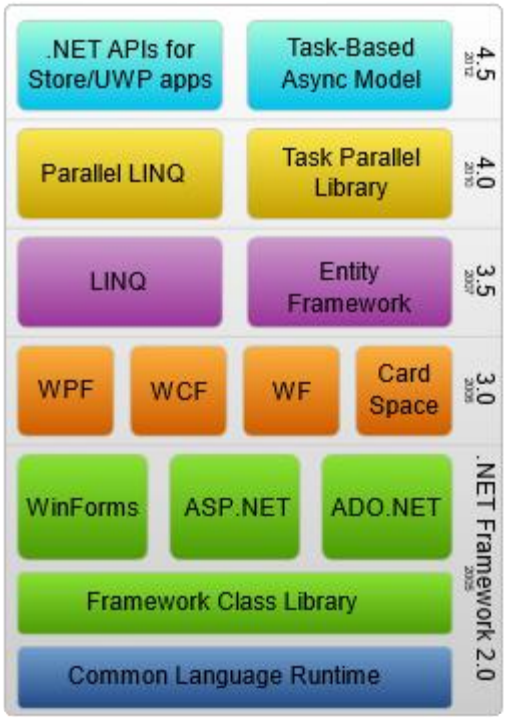
```
namespace MVC.API.Controllers
{
    [ApiController]
    [Route("/[controller]")]
    0 references | 0 changes | 0 authors, 0 changes
    public class ApiController : Controller
    {
        private DB_Entities _db = new DB_Entities();

        //[Authorize]
        [HttpGet("register")]
        0 references | 0 changes | 0 authors, 0 changes
        public ActionResult Register()...

        [HttpPost("register")]
        0 references | 0 changes | 0 authors, 0 changes
        public ActionResult Register(User _user)
        {
            if (ModelState.IsValid)
            {
                var check = _db.Users.FirstOrDefault(s => s.Email == _user.Email);
                if(check == null)
                {
                    _db.Configuration.ValidateOnSaveEnabled = false;
                    _db.Users.Add(_user);
                    _db.SaveChanges();
                    return RedirectToAction("Index");
                }
                else
                {
                    ViewBag.error = "Email already exists";
                    return View();
                }
            }
            return View();
        }
```

2. Services addControllers(), addControllersWithViews(), addRazorPages(), addMvc()

- https://dotnettutorials.net/lesson/difference-between-addmvc-and-addmvccore-method/

3. .NET Ecosystem: .Net / .Net Core / .Net Standard

- .NET Framework là 1 nền tảng lập trình và cũng là môi trường để thực thi ứng dụng trên Windows.



- 2 thành phần chính trong .NET Framework là Framework Class Library (FCL) và Common Language Runtime (CRL)
- FCL là các thư viện hỗ trợ sẵn, lập trình viên chỉ cần học các sử dụng và dùng lại để xây dựng app
- CLR như 1 sandbox là môi trường chạy ứng dụng. CLR còn hỗ trợ 1 số tính năng như manager security, manager memory, handler exception, garbage collection,…
- Khi compiler app, IDE sẽ dịch ra 1 ngôn ngữ trung gian - Microsoft Intermediate Language (MSIL). Khi run app, CLR sẽ dịch MSIL ra mã máy để thực thi.
- WinForms: windows desktop app
- ASP.NET: web app
- ADO.NET: connect DB (Oracle/MySQL)

- .NET Core là 1 framework độc lập với .NET nhưng được phát triển từ .NET, với mục đích có thể chạy trên đa nền tảng (Windows/Linux/MacOS…)
    - Không hỗ trợ winforms
    - CoreCLR
- .NET Standard là 1 bộ đặc tả kỹ thuật (specification) của những API chung trong các .NET framework. Có thể hiểu .NET Standard là dạng Interface còn .NET, .NET Core, Xamarin/Mono là các Implemention khác nhau
    - .NET, .NET Core chạy trên máy Desktop
    - Xamarin/Mono chạy trên Android/iOS

4. .NET SDK vs .NET Runtime

- .NET SDK: là những công cụ hỗ trợ cho việc build/run app (CLI, IDE Visual Studio)
- .NET Runtime: là những thứ cần cho việc run app (đã bao gồm trong .NET SDK)

5. Target framework

MVC.API.csproj

```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>netcoreapp5.0</TargetFramework>
  </PropertyGroup>
```

- Examples:
    - .NET Standard → netstandard2.1
    - .NET Framework → net48
    - .NET 5+ (and .NET Core) → netcoreapp3.1, net5.0/netcoreapp5.0

- https://docs.microsoft.com/en-us/dotnet/standard/frameworks