**JWT**

**1: Instalar os pacotes NuGet (para cada projeto):**

* package Microsoft.AspNetCore.Authentication.JwtBearer;

**1.2: Definir esquema de autenticação do JWT:**

*builder.Services.AddControllers();*

*builder.Services.AddAuthentication("Bearer").AddJwtBearer();*

**1.3: Gerar e gerenciar tokens usando a ferramenta:**

*cd(local arquivo) dotnet user-jwts*

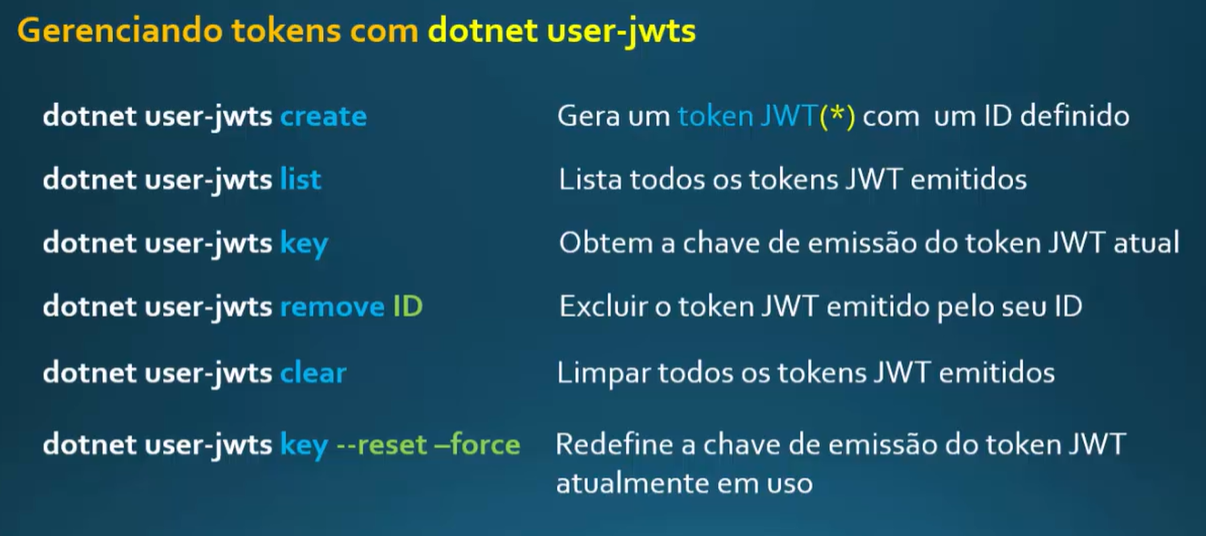
**1.4: Gerar token JWT pelo prompt de comando:**

*cd(local arquivo) dotnet user-jwts create*

**1.5: Proteger os endpoints com o atributo:**

**[Autorize]**

Comandos JWT:



**2: Habilitar o Identity:**

* package Microsoft.AspNetCore.Identity.EntityFrameworkcore;

**2.1: Alterar a classe de context para herdar de IdentityDbContext:**

*public class AplicacaoDbContext : IdentityDbContext*

**2.2: Configurar o Identity na classe program:**

*builder.Services.AddIdentity<IdentityUser, IdentityRole>() //perfis do identity*

*.AddEntityFrameworkStores<AplicacaoDbContext>() //relacionar os dados*

*.AddDefaultTokenProviders();* //lida com os tokens padrão

**2.3: Aplicar o Migrations do EF para gerar o BD:**

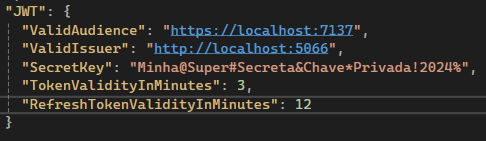
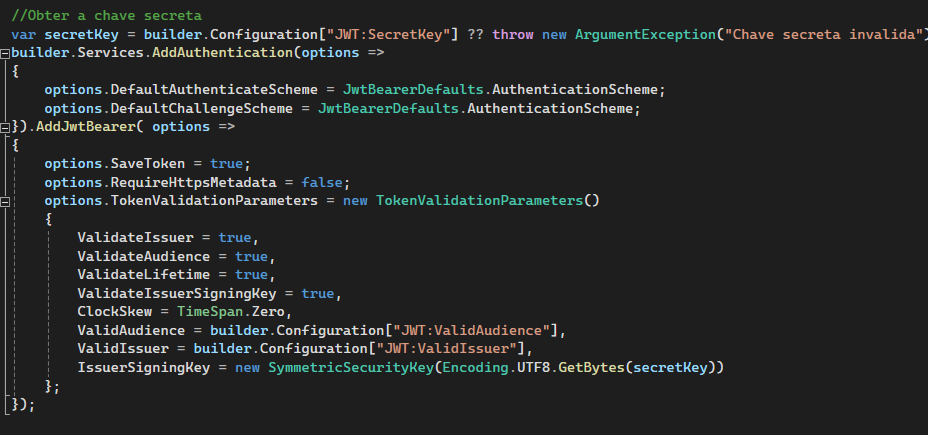
***Gerar o script da migração:***

* + - *dotnet ef migrations add* ‘CriarTabelaIdentity’ –verbose (para detalhar a migraçao)

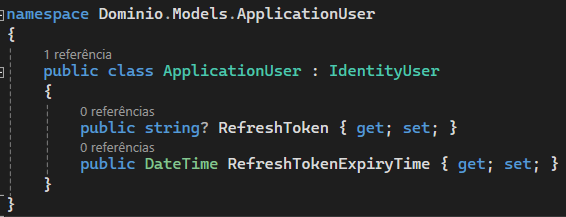
***Aplicar o script de migração:***

* + - *dotnet ef database update* –verbose

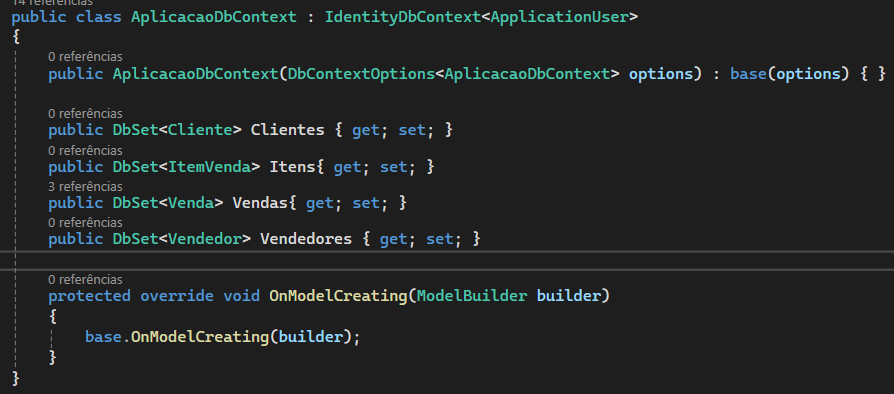
**3: Informações para gerar o token:**

* **Definir a sessão do JTW no appsetting.json:**
  + - SecretKey precisa ter um bom tamanho
  + ****
* **Habilitar e configurar autenticação do JWT Bearer no program:**
  + **

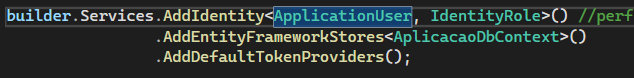
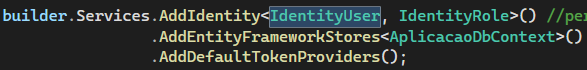
**4: Criar a classe ApplicationUser:**

* 

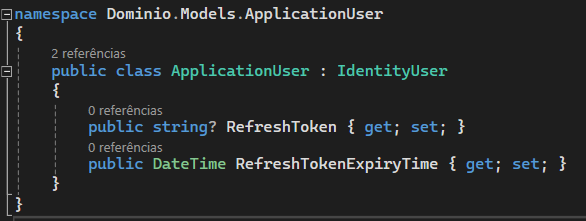
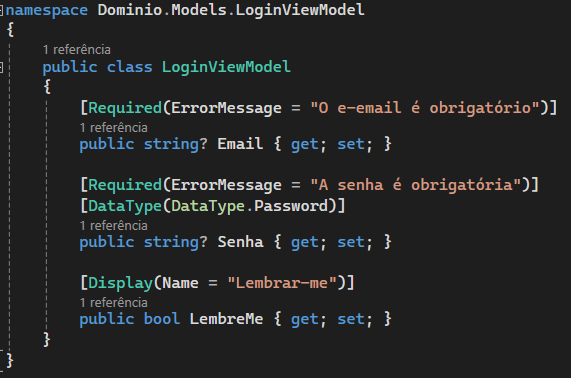
**4.1: add o ApplicationUser no DbContext:**

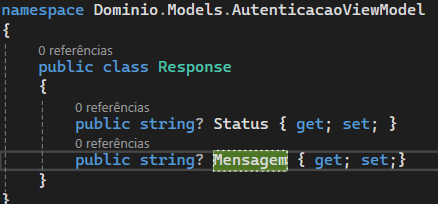
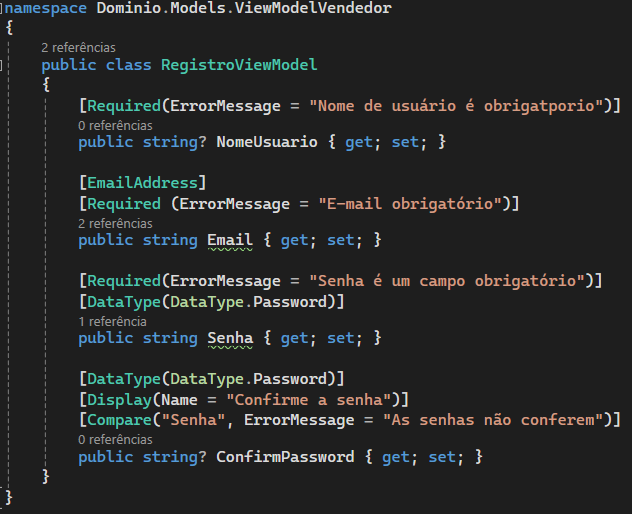


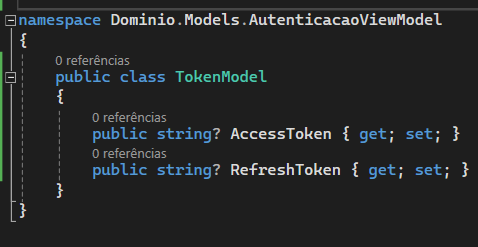
**4.2: No Program muda Identity por ApplicationUser:**



**56: Criar as classes da autenticação:**

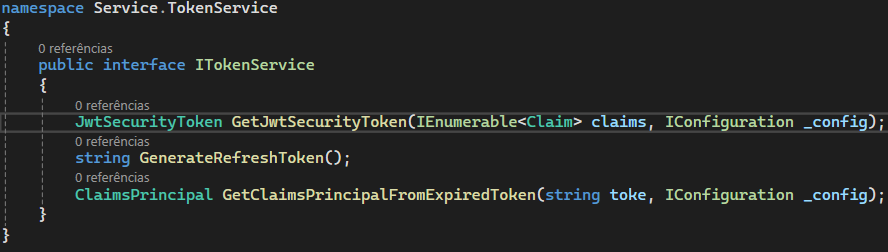
 





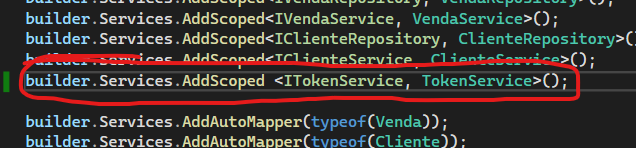
**7: Criar o serviço para gerar o Token e o RefreshToken**

* Criar a Interface ITokenService e a classe TokenService:

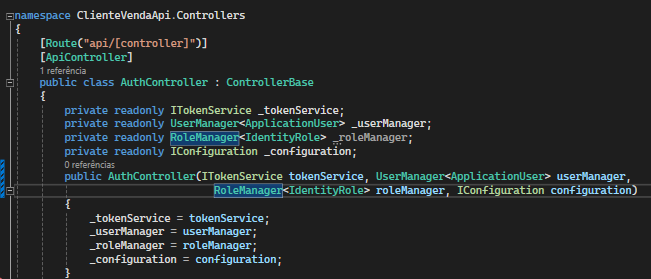
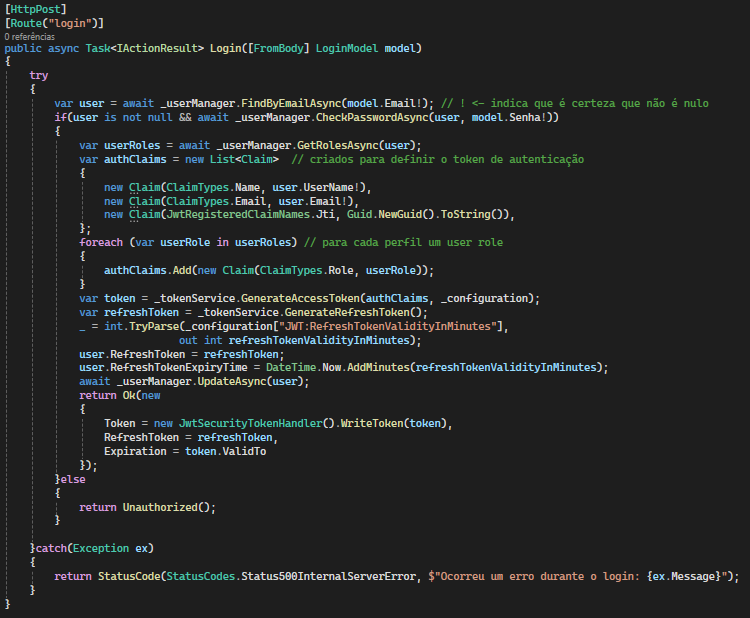
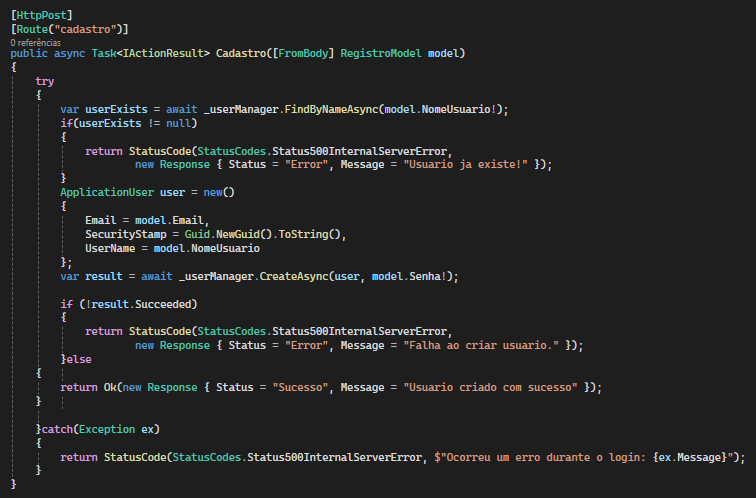
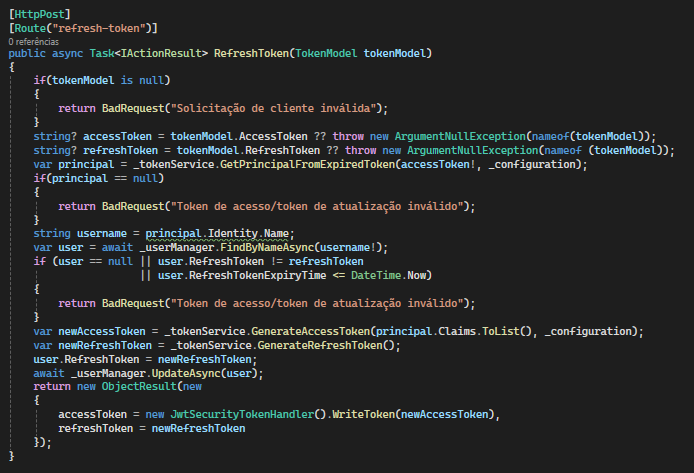
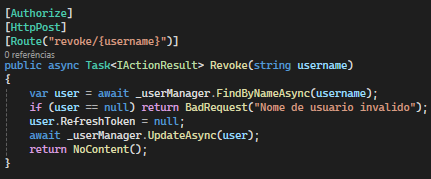


* Criar o service TokenService:
  + Registrar o service no program

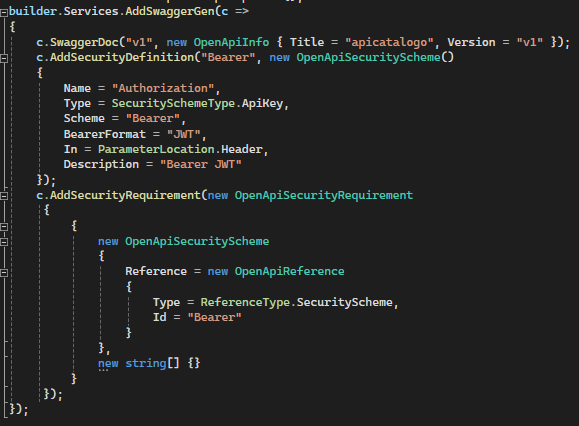




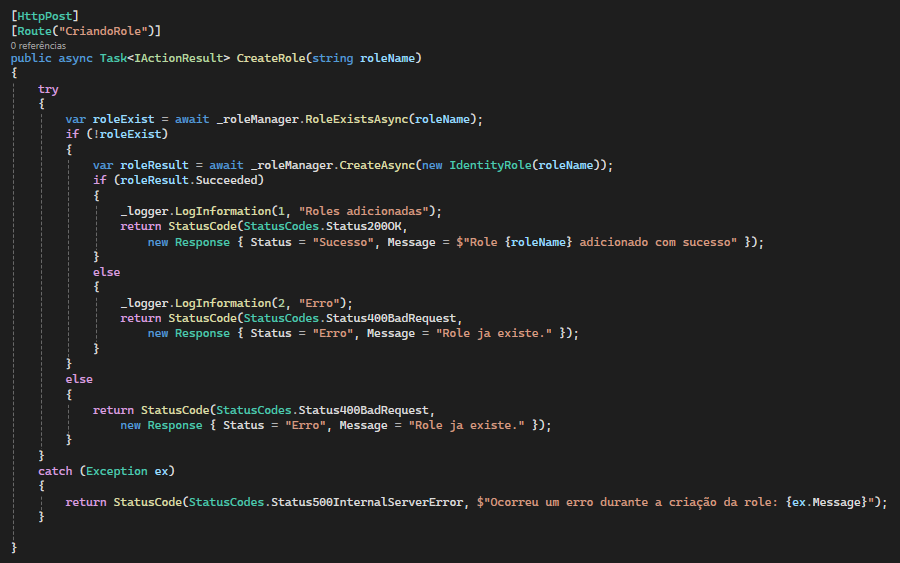
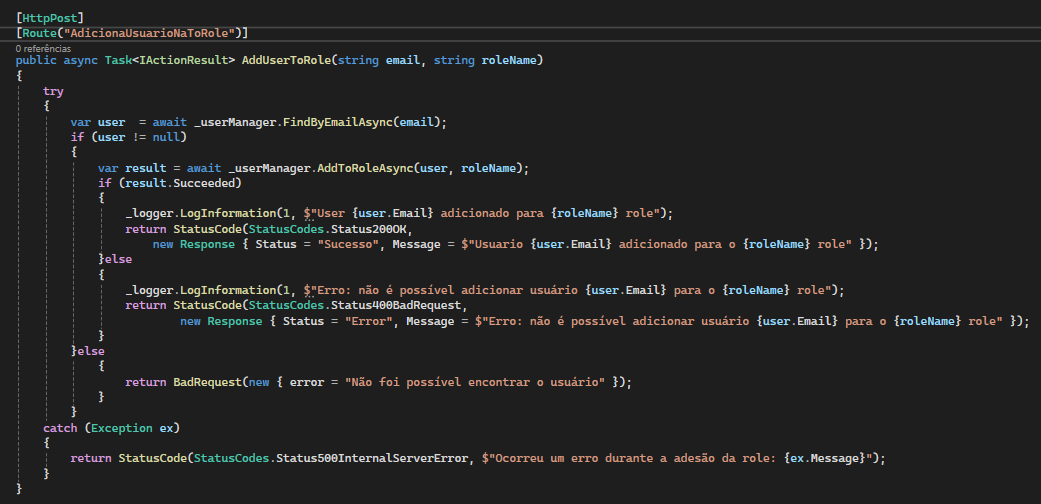
**8: Controlador authController:**

* Criar o controller AuthController com os endpoints;
  + Login;
  + Register;
  + RefreshToken;
  + Revoke
  + Injeção de dependencia no controller:
* 
  + Login
* 
  + Cadastro
* 
  + Refresh-token: Token de pós experaçao do token original
* 
  + Revoke: Para revogar o token
* 

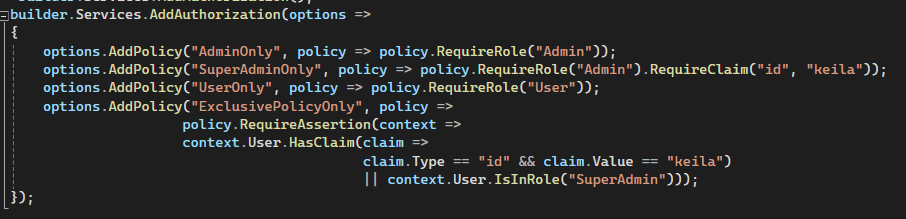
**9: Configurar o SwaggerGen para que tenha o cadeado de bloqueio por token:**

* Fazer os testes
* 

**10: Autorização baseada em Politica:**

* Criando um perfil(Role)
  + Incluir a instancia de Logger no controlador (injeta no construtor):
    - private readonly ILogger<AuthController> \_logger;
  + Criar o Endpoint com a lógica para criar uma role(ex: Admin)
    - 
  + Criar o EndPoint com a lógica para Adicionar usuarios a role:
    - 

**11: Configuração e criação das politicas de autorização no program:**

* RequiredRole
* RequiredClaim
* Required Assertion
  + 
* Adicionar o id no endpoint de Login na lista das Claims
  + 