- Refresh Tokens -

- what's Refresh Token ? -

- لما نعمل ال SignIn ال Server هيبدأ يـ generate ← 2Tokens

2Tokens ↙ Jwt Token ↘ Referesh token of any type [Jwt ---]

ليه فترة أطول بكتير من ال Token الأول

دا

- كدا ال Client يكون عنده ٤ معلومات ←
  - ال Token الأساسي
  - ال Token دا هيكون valid لمدة قد إيه
  - ال refereshToken
  - ال refereshToken دا هيـ Expire إمتى.

- لو هو لقى إن ال Token الأساسي اللي معاه خلاص expired فبتالي هيبعتله ال Server
وهيبدأ يبعت ال refereshToken لل Server وهيبدأ يـ validate ال Token بعدكدا
هيـ regenerate ← Token جديد، هيبعته لل Client

- Add Refresh Tokens Table -

① هنروح على ال Entities ونضيف Class جديد نسميه RefreshToken
وهنخليه [Owned]

```
[Owned]
Public class RefreshToken
{
  Public String Token {get; Set;} = String.Empty;
  Public DateTime ExpiresOn {get; Set;}
  Public DateTime CreatedOn {get; Set;} = DateTime.UtcNow;
  Public DateTime? RevokedOn {get; Set;}

  Public bool IsExpired => DateTime.UtcNow >= ExpiresOn;
  Public bool IsActive => RevokedOn is null && !IsExpired;
}
```

② هنروح على ال Nav. Prop في ال ApplicationUser ⇐

1:M

```
Public List<RefreshToken> RefreshTokens {get; Set;} = [];
```

③ هنروح على ال UserConfigurations ونكتب ال Config. بتاعت ال Nav. prop.

```
builder.OwnsMany(x => x.RefreshTokens)
       .ToTable("RefreshTokens")
       .WithOwner()
       .HasForeignKey("UserId");
```

④ آخر حاجة هنعمل Migration
و Update-Database

- Validate Jwt Tokens -

- المفروض هيكون عندنا EndPoint جديدة الـ EndPoint دي هتستقبل معلومتين :
أول واحدة ← الـ JWT Token اللي اليوزر بيستخدمها سواء كان Expired أو لا
تاني واحدة ← الـ RefreshToken

كدا هاخد الـ JWT Token وأعرف منه الـ UserId الموجود جوا الـ Claims اللي موجودة في الـ Token
بعدكدا هحتاج أروح الداتابيز أشوف الـ refreshToken دا بتاع User مين؟

بعد كدا المفروض هقارن بين الـ UserId بتاع الـ JwtToken والـ UserId بتاع الـ refreshToken
لو لقيتهم الاتنين واحد ← معناه انهم بتوع نفس اليوزر ← هرجع

generate ← JwtToken جديد
← RefreshToken جديد

① هنروح على الـ interface ← IJwtProvider

String? ~~validation~~ validateToken(String token);

② هنروح نعملها Implement في الـ Class ← JwtProvider

```
Public String? validateToken (String token)
{
   var tokenHandeler = new JwtSecurityTokenHandler();
   var SymmetricSecurityKey = new SymmetricSecurityKey (Encoding.UTF8
                                 . ~~GetBytes~~ GetBytes(options.key));
   try
   {
      tokenHandler.validateToken (token, new TokenValidationParameters
        {
           IssuerSigningKey = SymmetricSecurityKey,
           validateIssuerSigningKey = true,
           ValidateIssuer = false,
           validateAudience = false,
           ClockSkew = TimeSpan.Zero
        }, out SecurityToken validateToken);

      var jwtToken = (JwtSecurityToken)validatedToken;
      return jwtToken.Claims.First (x => x.Type == JwtRegisteredClaimNames
                                              .sub).Value;
   }
   catch { return null; }
}
```

شكل الـ outPut

عشان أجيب الـ claims اللي موجودة عندي في الـ jwt

Generate RefreshTokens -

هنحتاج نعدل على ال AuthService في ال Method اللي اسمها GetTokenAsyn - ونخليها
ت return ← RefreshToken وال ExpairyDate بتاعه

① ف الأول ← هنروح على ال AuthResponse ونضيف

```
String RefreshToken,
DateTime RefreshTokenExpiration
```

② هنرجع على ال AuthService
هنضيف PrivateField

```
Private readonly int _refreshTokenExpiryDays =14;
```

③ هنروح نكريت PrivateMethod اسمها GenerateRefreshToken

```
Private Static String GenerateRefreshToken()
{
 return  Convert.ToBase64String(RandomNumberGenerator.GetBytes(64));
}
```

④ هنروح لل Method اللي اسمها GetTokenAsync بعد الجزء اللي عملنا فيه Generate
لل JwtToken

```
var refreshToken = GenerateRefreshToken();
var refreshTokenExpiration = DateTime.UtcNow.AddDays(_refreshToken
                                                      ExpiryDays);

  // هنضيف الحاجات دي لل User بتاعنا
user.RefreshTokens.Add(new RefreshToken
 {
   Token = refreshToken,
   ExpiresOn = refreshTokenExpiration
 });

// هنسيف الكلام دا في الداتابيز عندي

await  _userManager.UpdateAsync(user);
```

هنضيف الداتا اللي كريتناها لل return

```
return new AuthResponse( ____ , refreshToken, refreshTokenExpiration);
```

- Add GetRefreshTokenMethod -

الوقتي بقا هنعمل ميثود وظيفتها إنها تـ generate Token جديد
باستخدام الـ JwtToken والـ RefreshToken اللي معانا
هنعمل الميثود دي في الـ AuthService

① هنروح على الـ IAuthService

```
Task<AuthResponse?> GetRefreshTokenAsync(string token, string refreshToken,
                     cancelationToken cancellationToken = default);
```

② هنروح نـ implement الميثود في الـ AuthService

```
Public async Task<AuthResponse?> GetRefreshTokenAsync(String token,
    String refreshToken, CancellationToken CancellationToken = default)
{
  var userId = _jwtProvider.ValidateToken(token);
  if (userId is null)
     return null;

  var user = await _userManager.FindByIdAsync(userId);
  if (user is null)
      return null;

  var userRefreshToken = User.RefreshTokens.SingleOrDefault(
        x => x.Token == refreshToken && x.IsActive);

  If (userRefreshToken is null)
    return null;

  userRefreshToken.RevokedOn = DateTime.UtcNow;
```

هنا دايمًا نـ generate Token جديد

```
  var (newToken, expiresIn) = _jwtProvider.GenerateToken(user);
  var newRefreshToken = GenerateRefreshToken();
  var refreshTokenExpiration = DateTime.UtcNow.AddDays(_refreshTokenExpiryDays

  user.RefreshTokens.Add(new RefreshToken
  {
   Token = newRefreshToken,
   ExpireOn = refreshTokenExpiration
  });
await _userManager.UpdateAsync(user);
return new AuthResponse ( user.Id, user.Email, user.FirstName,
    user.LastName , newToken, ExpiresIn, newRefreshToken,
    refreshTokenExpiration);
}
```

-Add RefreshToken End Point-

① هنروح على الـ Authentication ونضيف 2 records

RefreshTokenRequest   RFreshTokenRequest Validator

```
Public record RefreshTokenRequest
(
  String Token,
  String RefreshToken
);

Public class RefreshTokenRequestValidator : AbstractValidator<
                                   RefreshTokenRequest>
{
  Public RefreshTokenRequestValidator ()
  {
    RuleFor (x => x.Token).NotEmpty();
    RuleFor (x => x.RefreshToken).NotEmpty();
  }
}
```

② بعد كدا هنروح على الـ AuthController ونضيف الـ EndPoint

```
[HttpPost ("refresh")]
Public async Task<IActionResult> RefreshAsync (
        RefreshTokenRequest request, CancellationToken CancellationToken)
{
   var authResult = await _authService.GetRefreshTokenAsync (
                  request.Token, request.RefreshToken ,
                  cancellationToken );

  return authResult is null ? BadRequest ("InvalidToken")
                            : Ok (authResult);
}
```

RevokeRefreshToke-

[illegible] وعملها
علام [illegible] البروجكت

① هنروح على ال IAuthService

```
Task<bool> RevokeRefreshTokenAsync (String token, String refreshToken,
                CancellationToken cancellationtoken = default);
```

② هنروح على ال AuthService ونعملها implement

```
Public async Task<bool> RevokeRefreshTokenAsync (String token, String refreshToken,
                cancellationtoken cancellationToken = default)
{
  var userId = _jwtProvider.validateToken(token);
  if (userId is null)
       return false;

  var user = await _userManger.FindByIdAsync(userId);
  if (user is null)
       return false;

  var userRefreshToken = user.RefreshTokens.SingleOrDefault(
                x => x.Token == refreshToken && x.IsActive);

  if (userRefreshToken is null)
       return false;

  userRefreshToken.RevokedOn = DateTime.UtcNow;

  await _userManger.UpdateAsync(user);

  return true;
}
```

③ هنروح على ال AuthController

```
[HttpPost ("revoke-refresh-token")]
Public async Task<IActionResult> RevokeRefreshTokenAsync ([FromBody]
            RefreshTokenRequest request, CancellationToken ___)
{
var isRevoked = await _authService.RevokeRefreshTokenAsync(
            request.token, request.RefreshToken, Cancellationtoken);
return isRevoked ? Ok() : BadRequest ("operationfaild");
}
```