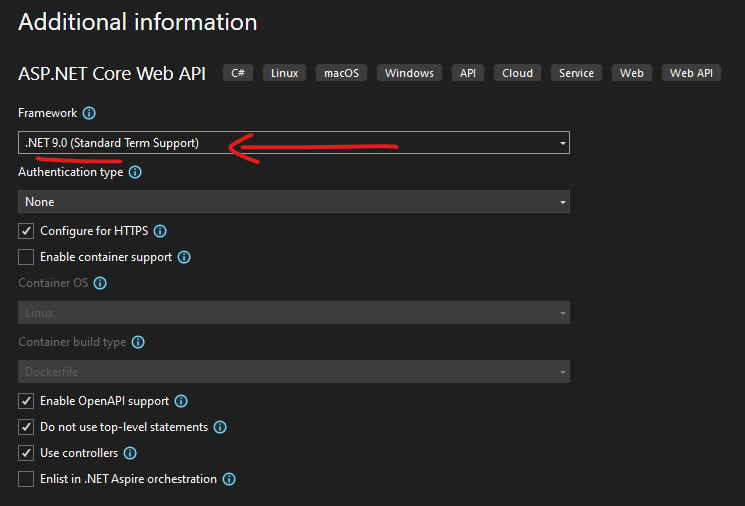
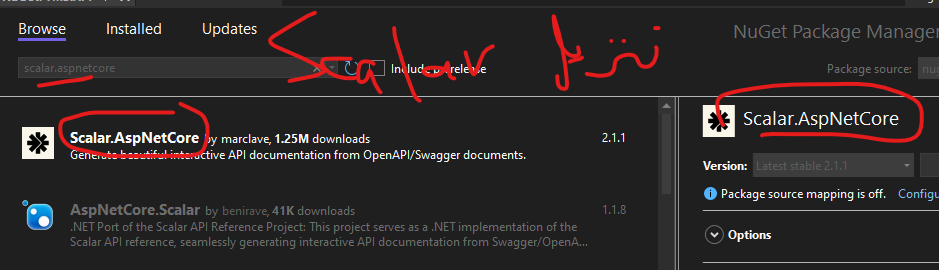
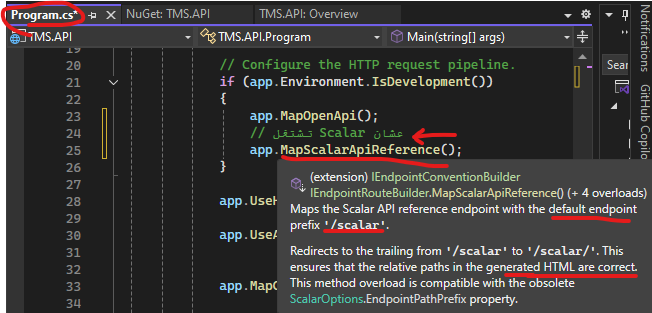
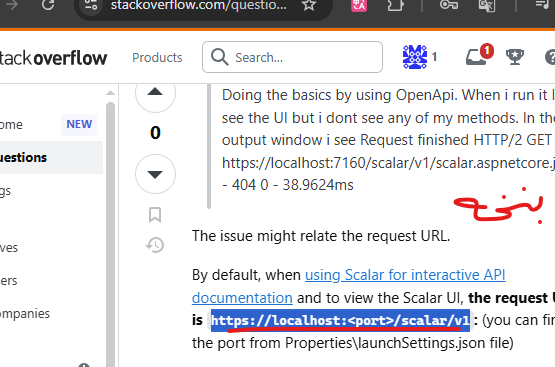


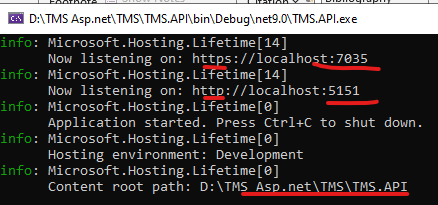
****  
**مكان واسم ال Solution وال Project.  
لاحظ ان اسم ال Solution غير ال Project.**

**.NET 9**

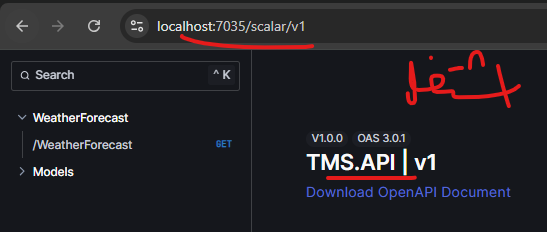
****

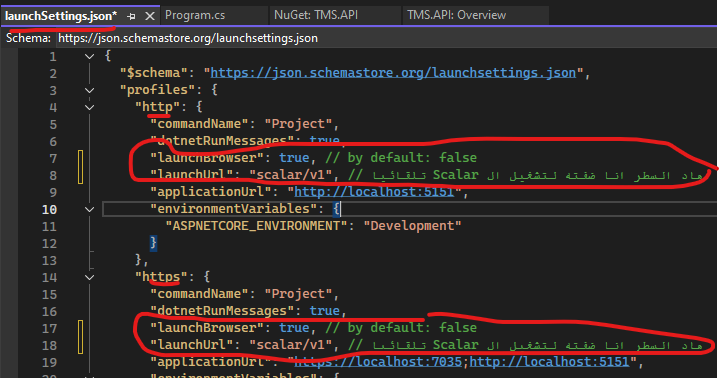
**تنزيل مكتبة Scalar.**

**منضيف app.MapScalarApiReference(); عشان تشتغل ال Scalar.**

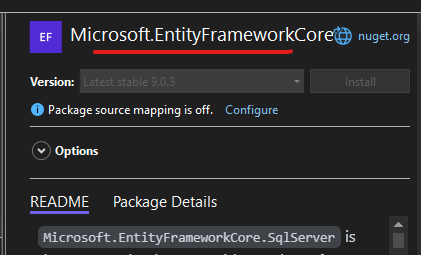
**  
  
بنسخ هاد الرابط وحسب ال port الي عندي بغيره لاله**

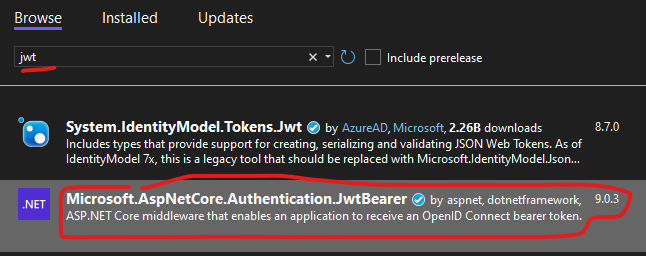
**اذا بغير ال port الى 7035 لل https و 5151 لل http.**

****

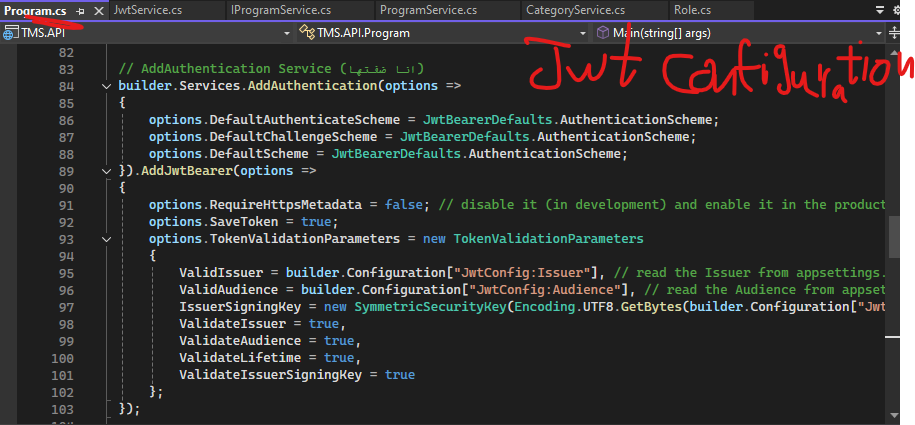
**  
اشتغل وكله تمام (يدويا نسخته ولصقته).**

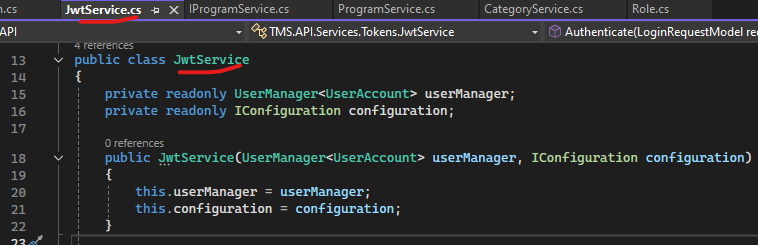
**بروح على launchSettings.json عشان هيك بصير يشتغل من حاله , مش انا كل مرة اروح يدويا واكتب الرابط كل ما بدي اعمل run!  
هسا هيك بمجرد ما اعمل run من حاله بفتح ال Scalar.**

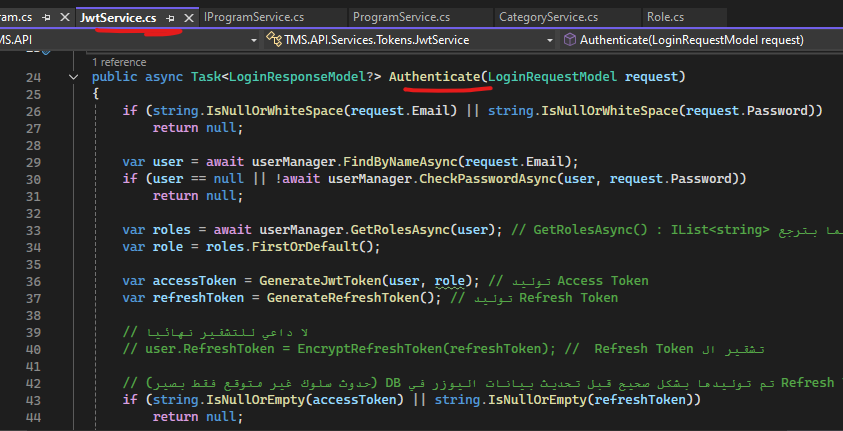
****

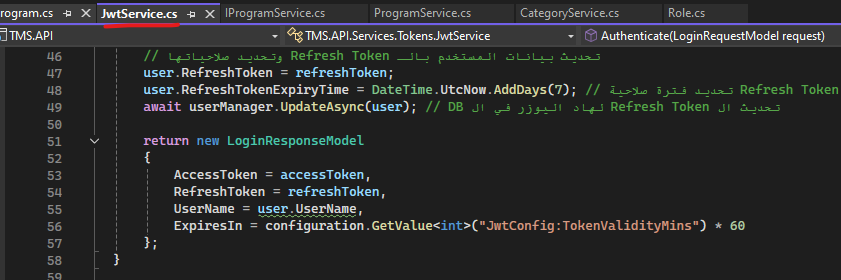
**مننزل المكتبتين Tools,SqlServer  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

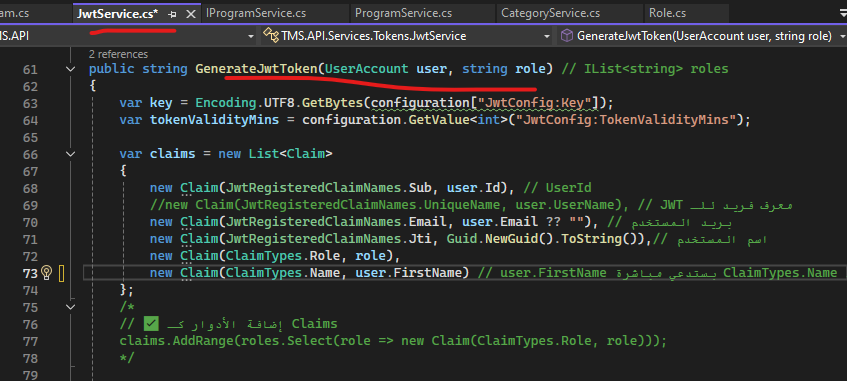
**مننزل ال JWT عشان موضوع التوكنز.**

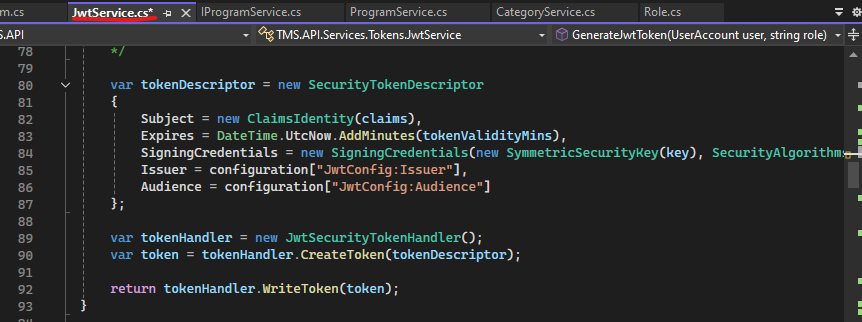
****

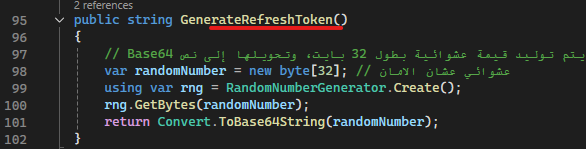
**ال configuration هاي ضرورية لل JWT**

****

****

****

****

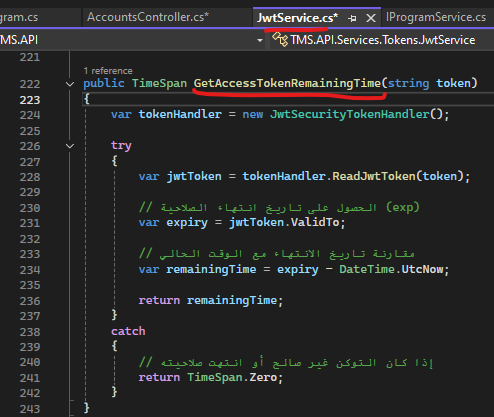
****

**Tokens**

**هاد الشغل الي رح نمشي عليه.**

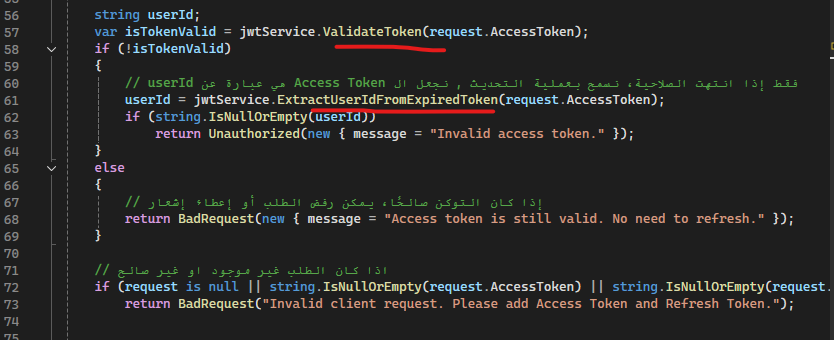
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

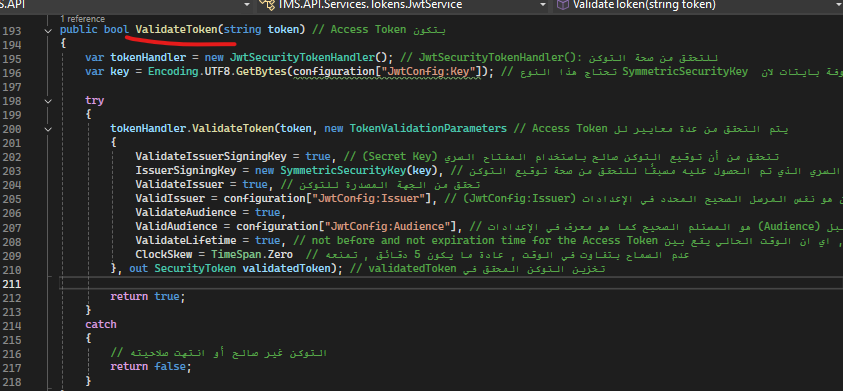
**الان في AccountController في ال Refresh Token**

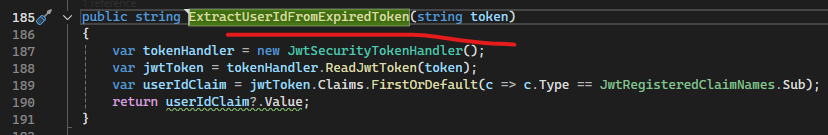
****

**بداية نفحص اذا ال Access Token خالصة ولا لا (المدة هي 30 د) بعد هيك**

**بتكون خالصة , ميثود ال GetAccessTokenRemainingTime بفحص.  
اول اشي بجيب تاريخ التوكن الحالي وبطرحه من التاريخ الحالي , اذا كان اكبر من صفر اذا هي لسا مش خالصة , غير هيك اذا هي خالصة ممنوع يعمل ريفرش  
(طبعا هاد الكود فقط في مرحلة ال Development بفيد للتتبع لا اكثر)**

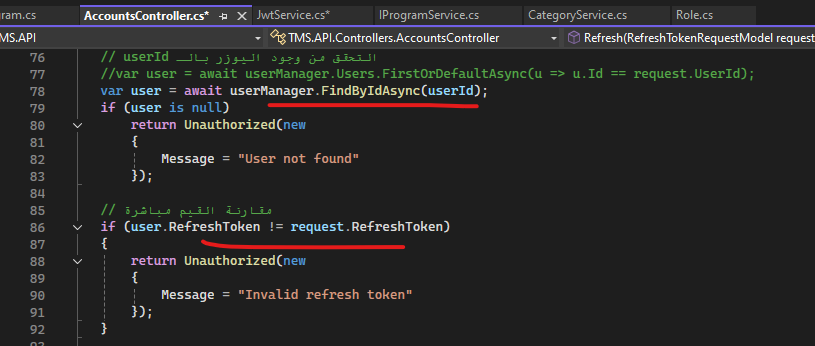
****

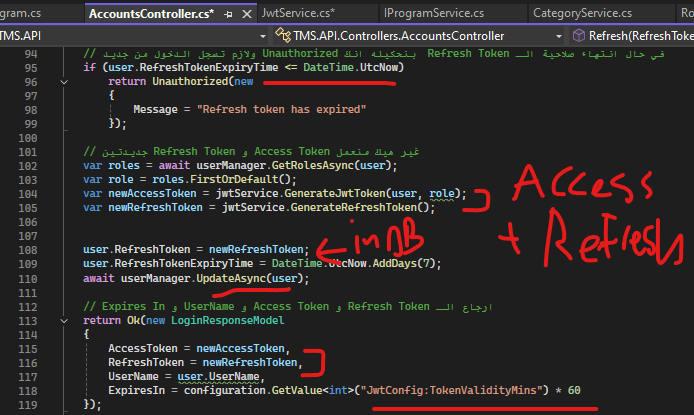
****

****

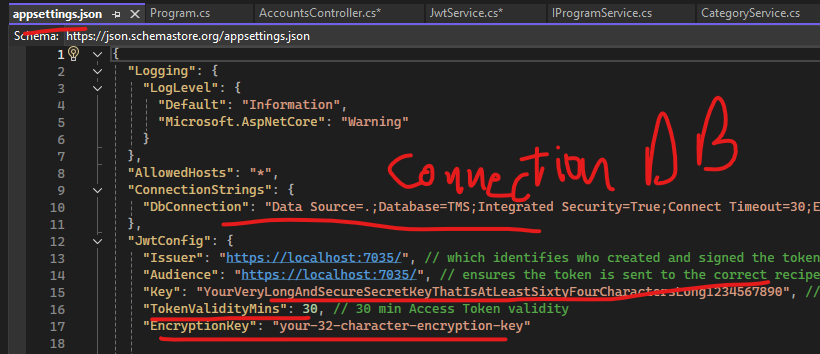
**منفحص اذا ال Access Token صالحة ولا لا , اذا مش صالحة معناها بقدر بهاي المرحلة اعمل Refresh Token لكن بالاول بدي افحص انو ال Access Token القديمة ما تكون null او empty , اذا هيك فهي invalid اذا مش هيك فهي ما خلصت مدتها ولا داعي لعمل Refresh**

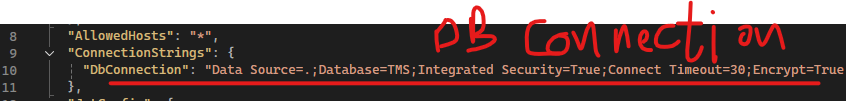
**الان بنفحص اذا ال request نفسه كان null او ال Access Token او ال Refresh Token null او Empty , اذا اه معناها لازم يدخلهن ال 2 مع بعض.**

****

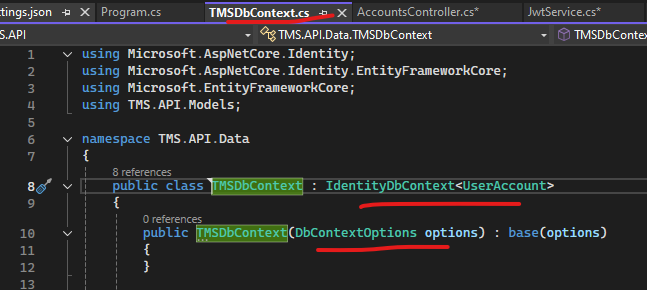
**  
الان بنفحص هاد ال user اذا موجود في DB , اذا مش موجود معناها بكون ب null , اذا موجود بدي اقارن ال RefreshToken تاعته اشوف اذا مش نفسها في ال DB اذا مش نفسها فال Refresh الي دخلها غلط! ممنوع يعمل Refresh اذا.**

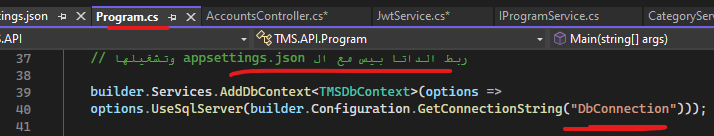
**الان ضمنا انو ال Refresh الي دخلها نفس الي في DB.  
الان بنفحص اذا ال Refresh خلصت او لا , اذا خلصت معناها بقدرش يحدثها  
اذا ما خلصت , بنجيب ال Role لهاد ال user (لانو احنا عالاكيد منكون مدخلين فقط Role واحدة لكل يوزر) ومننشئ Refresh + Access جديدة ومنحدث ومنعكس ال Refresh في DB ومنضيف لل Refresh 7 أيام للانتهاء ومنحدث ال DB , ومنرجع Ok بالبيانات الي قدامك بالإضافة لوينتا بتنتهي (منحول من دقائق الى ثواني)  
ملاحظة : ال GetRolesAsync() برجع IList دائما , اذا بحوله عشكل عنصر.**

****

**  
طبعا هون ببين ال JWT Config الي احنا عاملينه , لاحظ ال Key, TokenValidityMins عشكل دقائق بتكون نص ساعة 30 د  
وكذلك ال EncryptionKey الي عنا  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

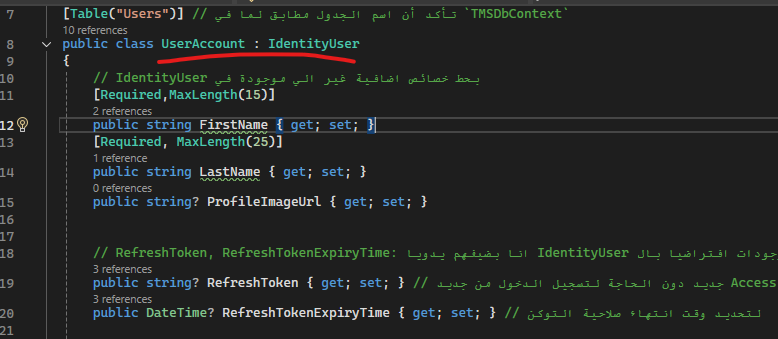
**الاتصال مع ال DB.**

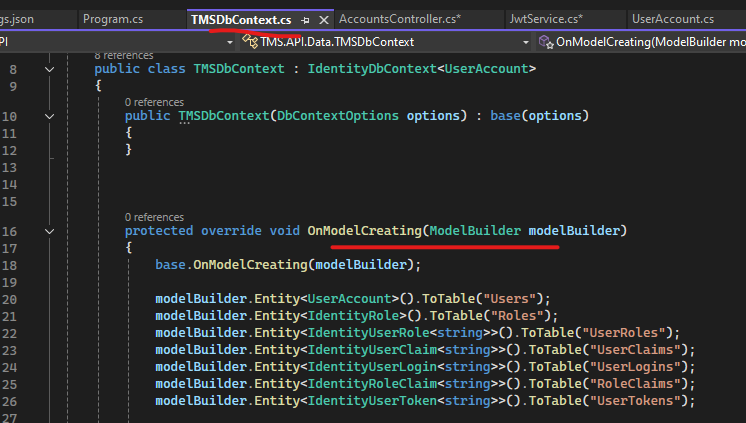
****

**هاد ال TMSDbContext للاتصال مع ال DB بطريقة ال DI**

**ربط ال DB مع ال appsettings.json.**

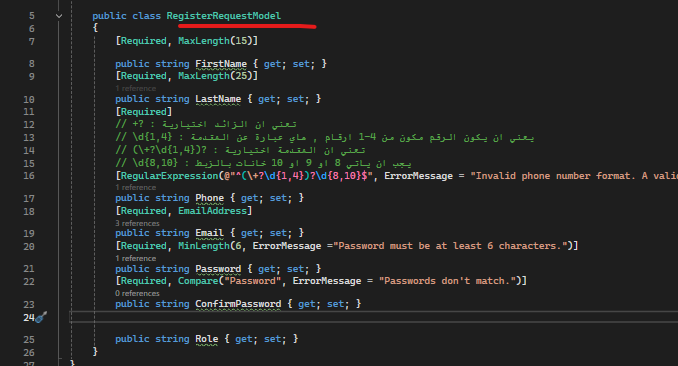
**لاحظ أصلا ان ال TMSDbContext وارث من ال IdentityDbContext<UserAccount>**

****

**  
  
ال UserAccount بورث من ال Identity وبضيف عليها كم Properties جداد (الي قدامنا).**

**طبعا هون انا بس بعدل أسماء الجداول الي موجودة باي ديفولت في ال Identity الى هذه الأسماء بالصورة.**

**الان انا بدي اتتبع عمل ال Actions في ال AccountsController وحدة وحدة:**

1. **Register**

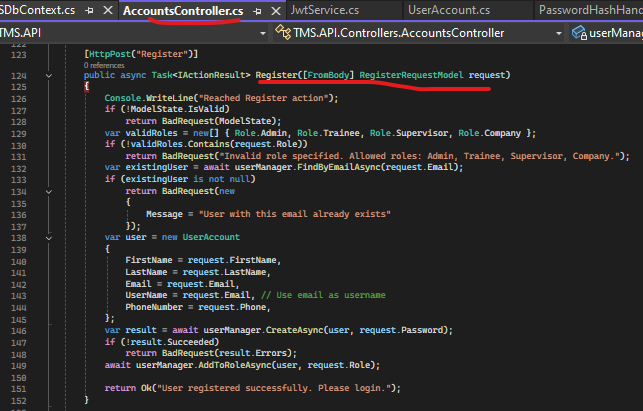
**هذا ال RegisterRequestModel الي اليوزر بدو يدخل فيه داتا عشان نعمل Register قدام.**

**التحقق من صيغة ال Phone:**

**التحقق من صيغة ال Email : لاحظ حكيناله انه [EmailAddress] هيك منجبره يدخل عشكل ايميل** [**ex@gmail.com**](mailto:ex@gmail.com)**) بهذا الشكل اجباري (لازم يكون في @)**

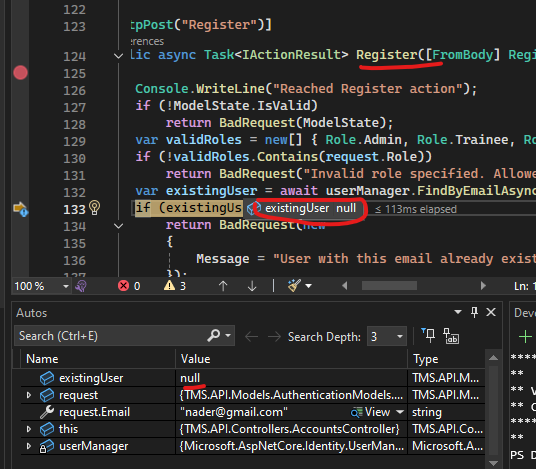
**لاحظ أيضا في ConfirmPassword : نتحقق منها مع ال Password الاصلية من خلال [Compaer()]**

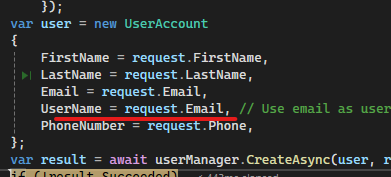
**ال Role اجباري تكون احدى ال 4 شغلات  
 [Admin, Trainee, Company, Supervisor] ورح يتم إيضاح لاحقا كيف.**

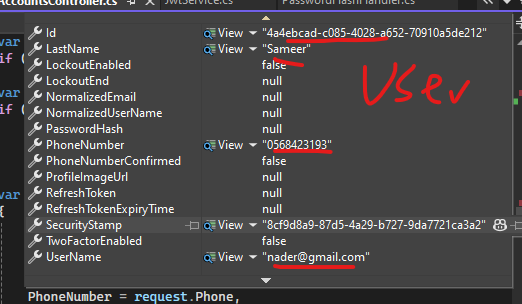
****

**سطر 127: منعمل Backend Validation بكون من خلال ال Data Annotation الي عملناه من قليل.  
سطر 129: منعرف Array تحتوي الأدوار المطلوبة فقط التي ممكن ان تكون.  
سطر 130: منتاكد اذا ال Role الي دخلها اليوزر احدى عناصر هاي ال Array ولا لا , اذا لا منعطيه BadRequest() غير هيك منكمل  
سطر 132: منجيب هاد اليوزر من ال DB بواسطة ال Email , اذا كان هاد اليوزر not null معناها هاد الايميل موجود من قبل ممنوع تسجل بالموقع بنفس الايميل! غير هيك منكمل (الايميل غير مستخدم من قبل)  
سطر 138: منعمل UserAccount جديد بالمعلومات الي بالصورة  
سطر 146: مننشئ هاد اليوزر مع ال password الي دخلها وبتكون بال DB لانو ال CreatAsync بكون تابع فنكشن جوا ال Identity وهيك التصرف الافتراضي الها  
سطر 147: منفحص اذا لم تتم العملية بنجاح BadRequest() غير هيك منضيف ال Role الي دخلها لهاد اليوزر  
سطر 151: في النهاية منرجع Ok() وانو العملية تمت بنجاح.**

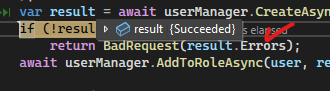
**التتبع :**

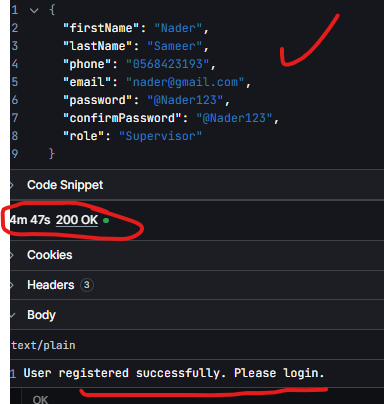
****

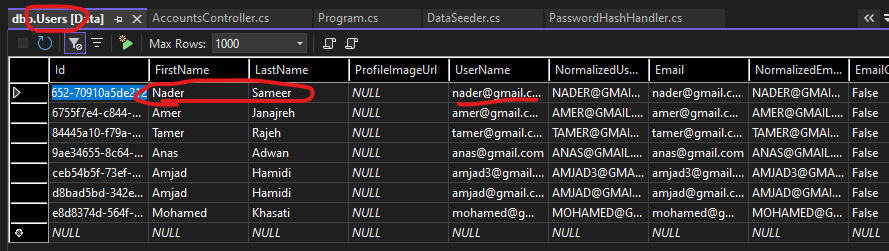
**  
لاحظ انه null معناها الايميل هاد غير موجود من قبل , اذا استطيع اعمل Register.**

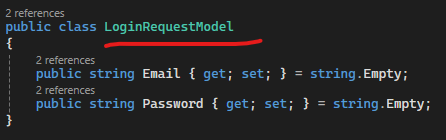
****

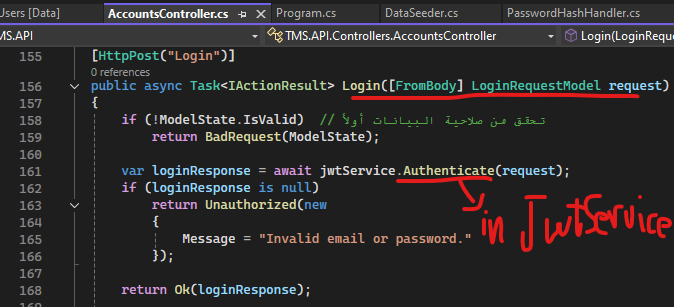
**لاحظ ان ال UserName بكون نفسه ال Email.**

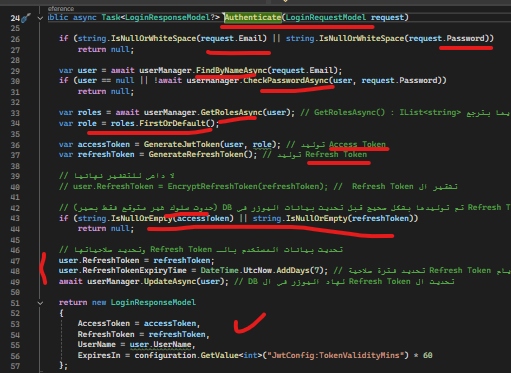
****

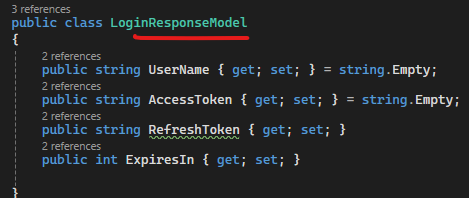
****

****

1. **Login**

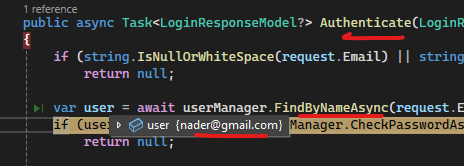
****

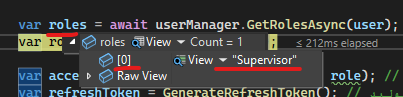
****

**  
في ال Login يتم التحقق من الباسورد والايميل بداية في ال Authentication ثم نحضر ال Role لهاد اليوزر , وننشئ Access + Refresh ثم نتحقق انه تم توليدهن بشكل صحيح , ثم نحدث ال Refresh الي مخزنة في DB في كلاس ال UserAccount , ثم نضيف Expiration لل Refresh بمقدار 7 أيام ونحدث ال DB ثم نرجع اوبجكت جديد يحمل ال LoginResponse**

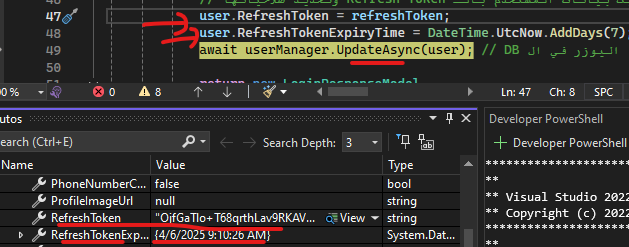
**تتبع ال Login:**

****

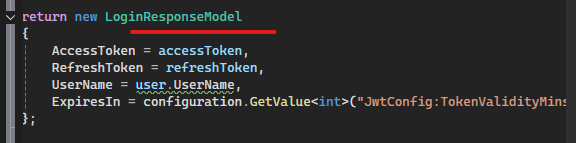
****

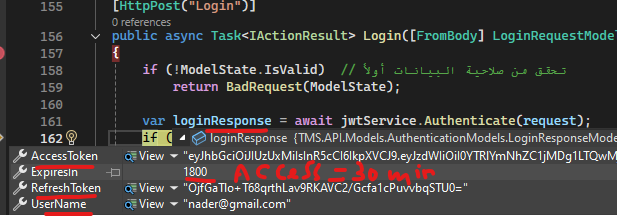
**لاحظ انه الايميل موجود اذا ال Authenticate لهون تمام , بدنا نفحص الباسورد ونكمل   
ثم نحضر ال Role لليوزر هاد.**

**  
لاحظ انه ال GetRolesAsync() بترجع IList مش عنصر واحد لذلك تحتها عملت** var role = roles.FirstOrDefault(); **عشان اجيبها عشكل single مش IList**

**ثم ننشئ Access + Refresh**

**بعدها منخزن ال Refresh في DB ثم منخزن  
ال Expire في ال DB منعطيها 7 أيام  
ومنحفظ الشغل هاد كله في DB.**

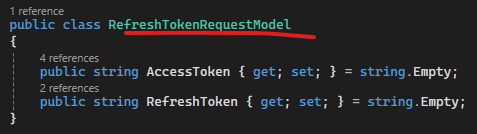
****

**ثم منرجع ال LoginRespone لل Login Action**

**  
لاحظ ال Expires بتنتهي بعد 1800 ثانية يعني 30د مثل ما برمجناها في ال JwtConfig في ال appsettings.json**

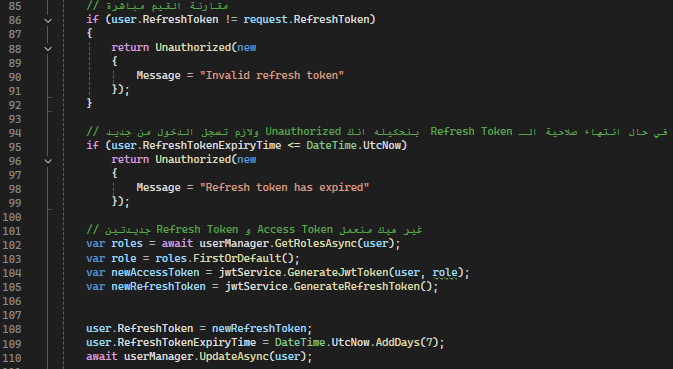
****

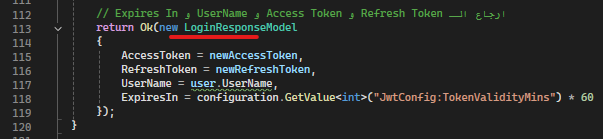
**تمت عملية ال Login كاملة.**

1. **Refresh**

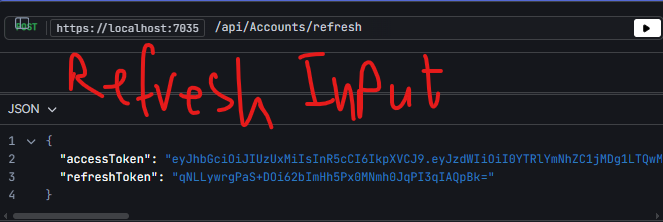
****

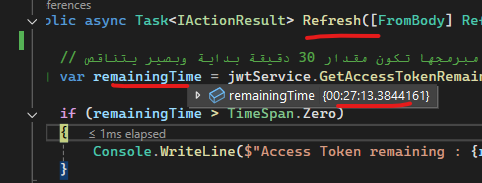
****

****

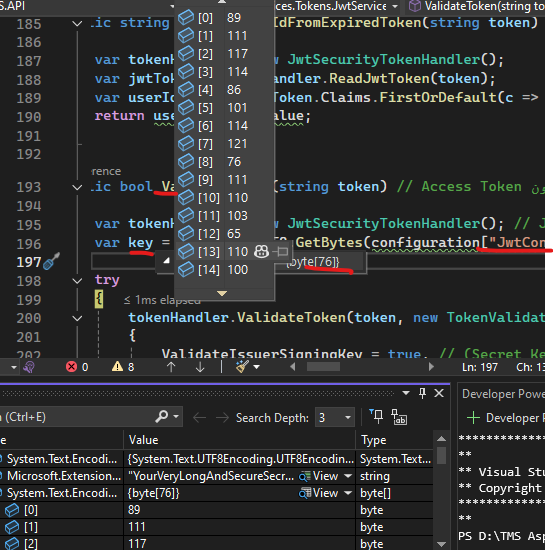
****

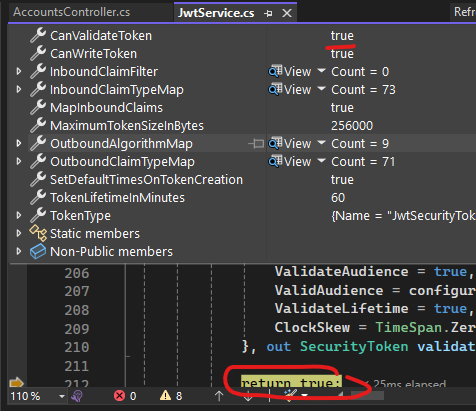
**تتبع ال Refresh:**

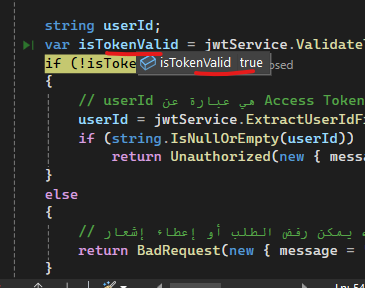
****

****

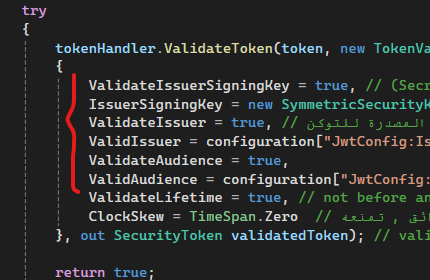
**معناها مرقت 3 دقايق وضايل لل Access Token 27 دقيقة**

**  
الان رح نتحقق من ال Access Token (Validation) غير موضوع ال time الي فحصناه بالمرحلة السابقة:**

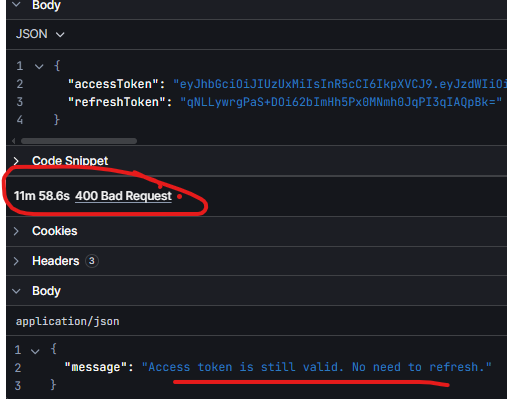
**  
لاحظ ان ال Key الي كتبناه بال JwtConfig في ال appsettings.json رجعلي طوله 76 بايت , وكل بايت بحتوي رقم معين (لاحظ انو البايت دايما رينجه 0-127) ما رح نلاقي اندكس منهن معين بحتوي اكثر من هيك.**

**تم التحقق من ال Access Token شغال بنجاح (طبعا تذكير : احنا تحققنا من الأمور الأخرى غير موضوع التايم الي فحصناه قبل)**

**طالما انه True التحقق من ال Access Token اذا هي مش خالصة اذا برجع BadRequest() لانو بديش اغيرها اذا وقتها مش خالص.**

**(يعني باختصار : انا مش رح اغير ال Refresh + Access الا اذا ال Access وقتها خالص او احد العوامل الأخرى**

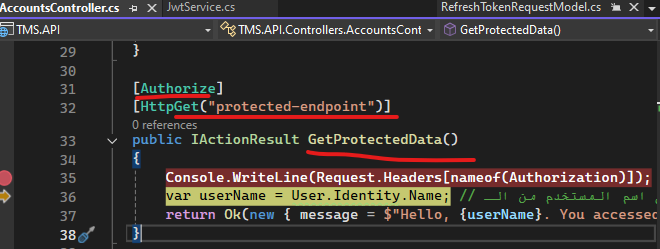
**لل Access الي هني هدول خالص)**

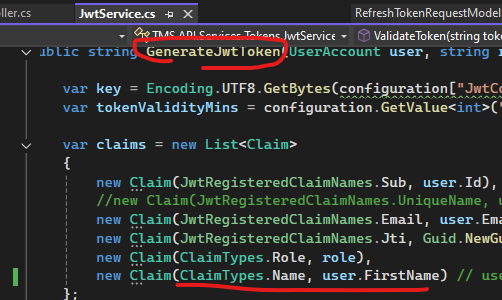
****

**اذا عشان ال Access وقتها ما خلص , اذا ما بقدر اغير واعمل  
Access + Refresh جديدتين.**

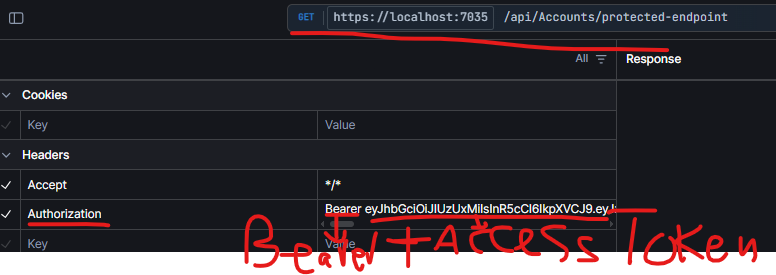
**انتهت عملية التتبع لل Refresh.**

1. **GetProtectedData**

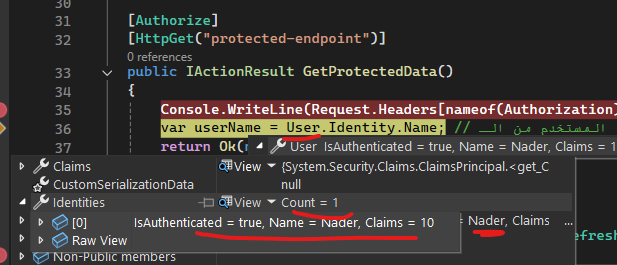
****

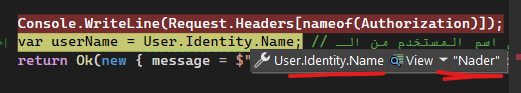
**ال User.Identity بتيجي لجلب ال Claim الموجودة في JwtService في فنكشن ال GenerateJwtToken   
User.Identity.Name تعني روح على ال Claim لل Name الي الها وشوف شو بقابلها وجيبه , الي هي هون (user.FirstName)**

**تتبع ال GetProtectedData :**

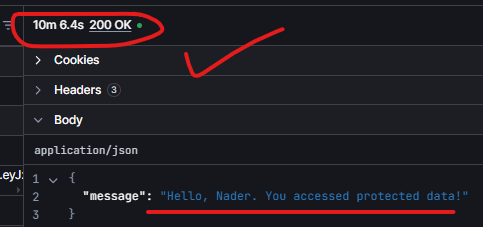


**منحط كلمة Authorization ك Key وال Value عبارة عن   
(Bearer + Access Token) الي حصلت عليها من Response ال Login**

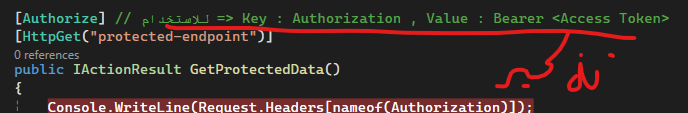
****

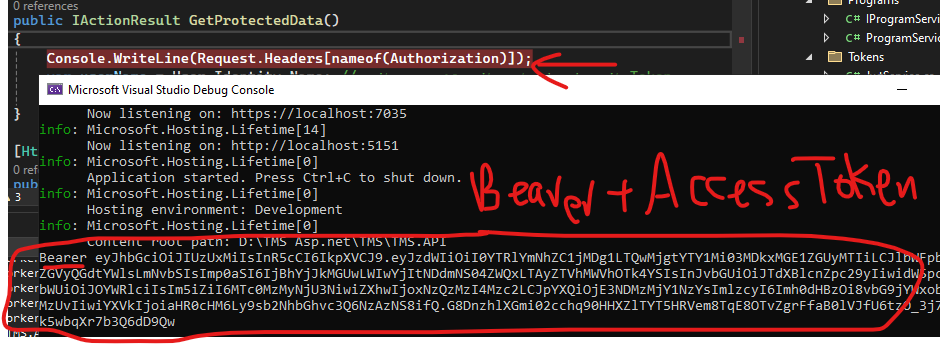
**  
ال User هون الها 1 Identity الي هي ل Nader الي دخلنا منه (التوكن الي فتنا من خلالها) , الان هاد ال User اله 10 Claims (الي شفناهن موجودات في ال JwtService في فنكشن ال GenerateJwtToken بال Claims الي موجودة.**

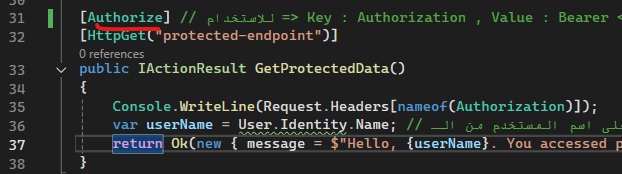
**لاحظ انه وصل لل user.FirstName الي مقابلة لل ClaimTypes.Name في ال Claims في ال GenerateJwtToken**

****

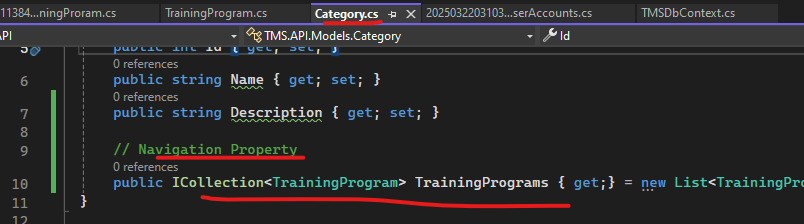
**تمت عملية التتبع لل GetProtectedData().**

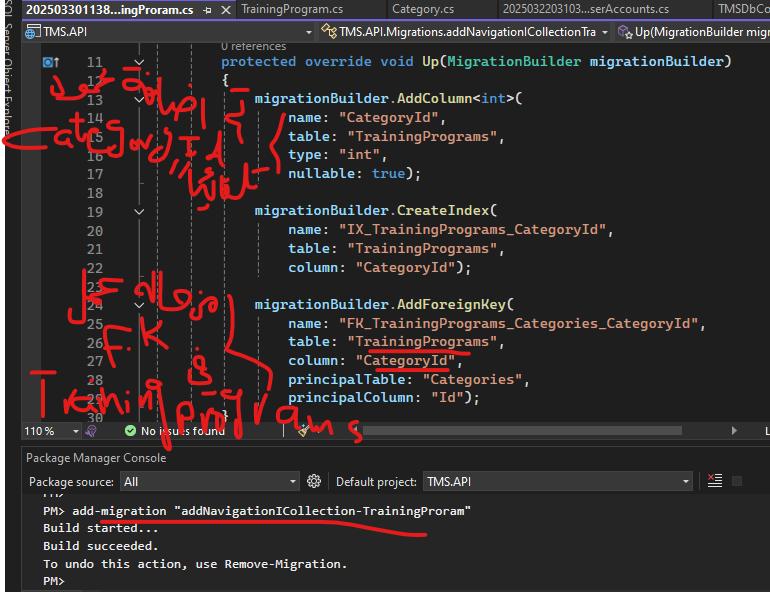
****

****

****

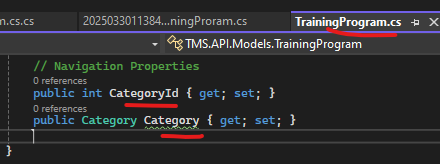
**تذكر : أي اشي فيه [Authorize] السلوك الافتراضي اذا ما تم التحقق بشكل صحيح بواسطة   
Key: Authorization, Value: Bearer + Access Token   
برجعله 401 Unauthorized**

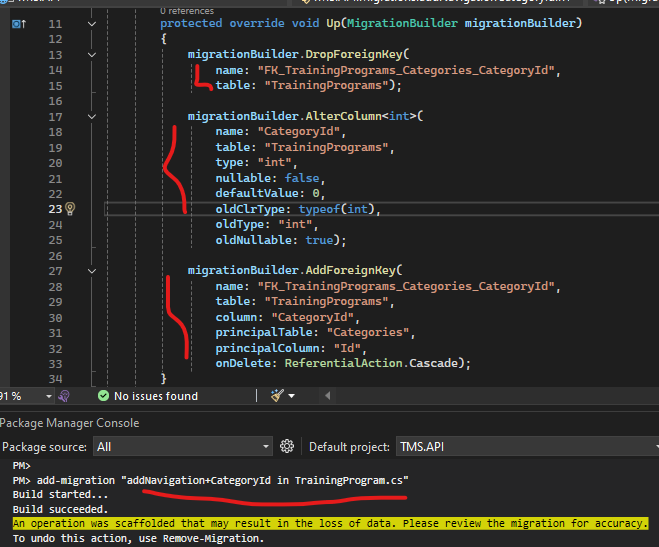
****

**إضافة Navigation Property في Category.cs لتؤشر على TrainingPrograms**

**تاثير ال Navigation الي اضفناها في Category.cs لتؤشر على TrainingPrograms:  
هاي ال migration الناتجة الها (تلقائيا لاحظ اضاف CategoryId في TrainingPrograms وجعله ك F.K) لكن اعطى ال F.K اسم غير ال CategoryId التقليدي (ضفله كلمات واكسبرشن معين)**

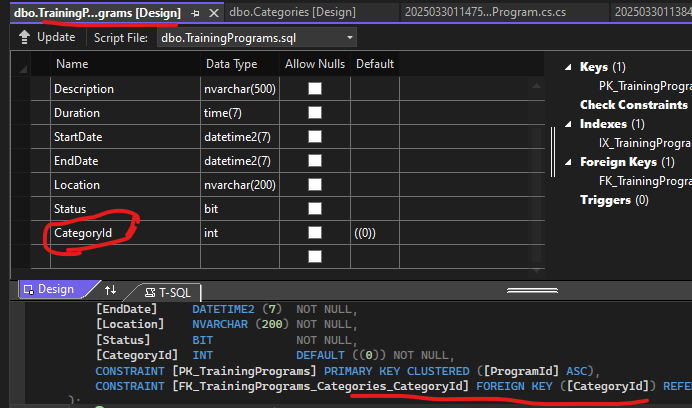
**ملاحظة : عملنا ICollection عشان ال List وكلهن بورثن منها واستخدمنا get; فقط بدون set; لاني ما بدي اعدل عليها فقط بدي استخدمها للعرض.**

****

****

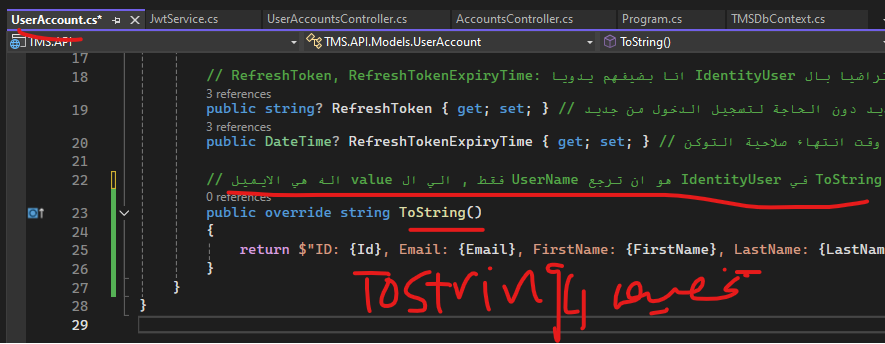
**  
لاحظ: إضافة ال CategoryId وال Navigation يدويا في ال TrainingProgram.cs ما اثر اشي عن ال migration الي قبلها , الا انو غيرت اسم ال F.K من الاسم القديم الى اسم CategoryId حاف (الي كتبناه الان)**

**ثم نعمل update-database عشان تصير هناك علاقة بين ال   
Category, TrainingProgram**

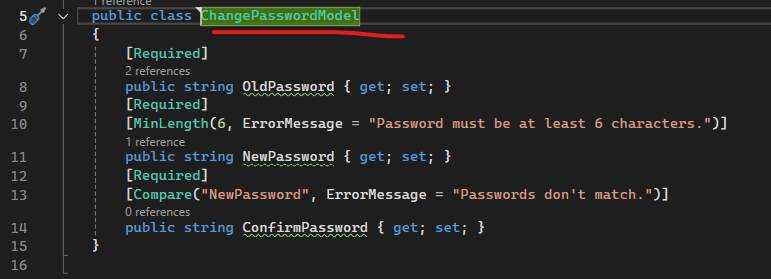
****

**تم انشاء العلاقة بنجاح (Category 1 – TrainingProgram many)**

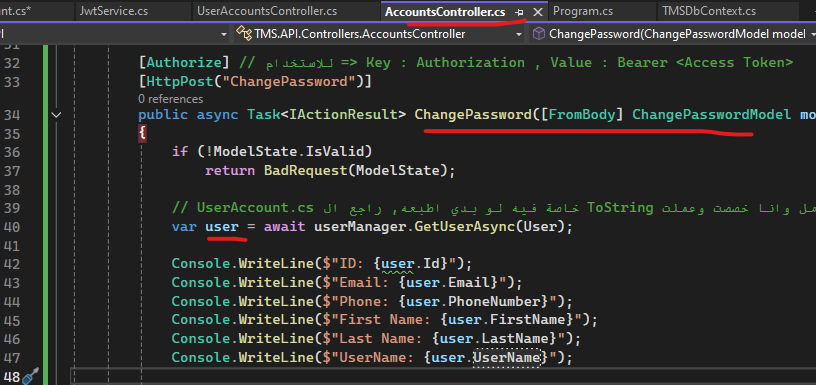
**عمل Change Password()**

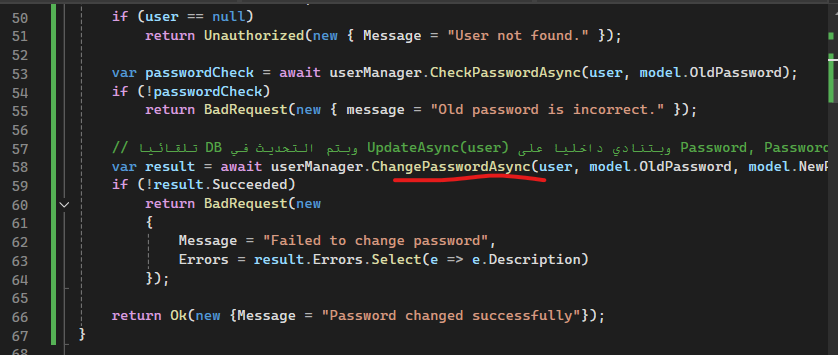
****

**السلوك الافتراضي لل UserIdentity انو يرجع في ال ToString ال UserName فقط (بحالتي ال UserName الفاليو اله الايميل) , لذلك غيرت فقط الشكل, طبعا مع عدم التاثير على إمكانية الوصول لل Properties.**

****

**هذا ال model الي عبارة عن request**

****

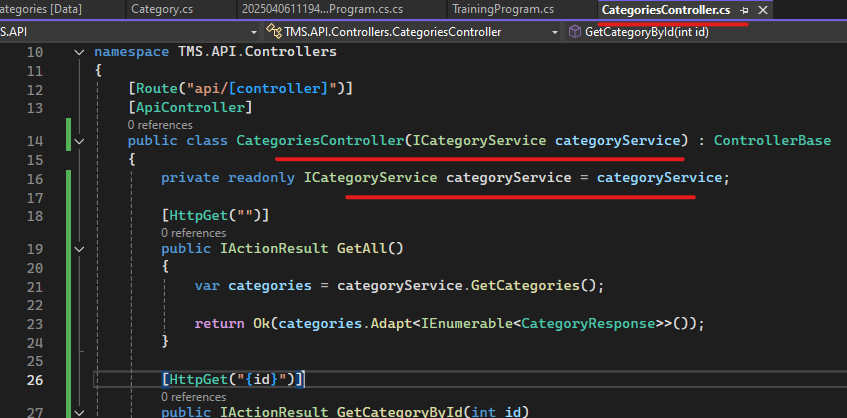
****

**في البداية لاحظ انه [Authorize] اذا بندخل ال  
 Authorization (Bearer + Access Token) وال Access Token منوخدها من ال Login**

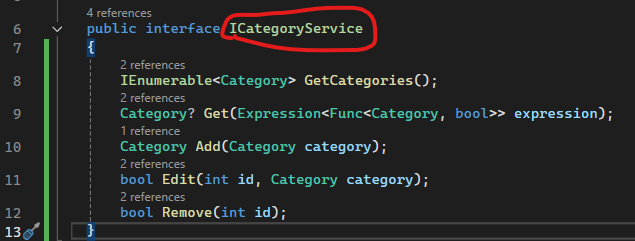
**في البداية منتحقق من الداتا الي مدخلة هل تطابق ال Valiadation باستخدام ال Data Annotation باستخدام ال ModelState  
ثم منجيب ال User كاوبجكت باستخدام ال GetUserAsync , ال ASP من حالها بتفك ال Access Token المبعوث وبتحوله لكائن اسمه User من نوع ClaimsPrincipal وفيه كل ال claims الموجودين (name,email,role,sub)   
تلقائيا بس اكتب**  GetUserAsync(User); **من حاله بروح لل sub الموجودة في JWTService.cs في ال Claims وبدور في DB على اليوزر الي عنده Id=sub.value يعني باختصار بجيب اليوزر من خلال ال Id**

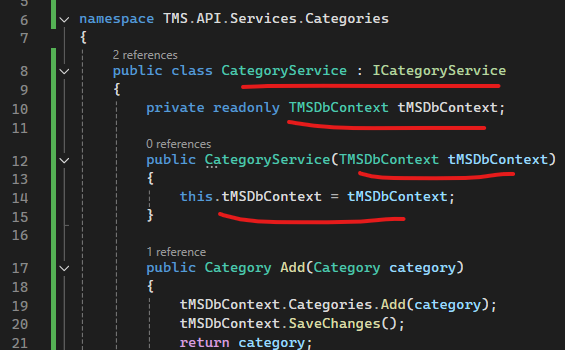
**الان منفحص اذا اليوزر فاضي منوقف منعطيه Unauthorized   
اذا لا منكمل ومنفحص ال Old Password مع الي موجودة في DB هل تطابقها فعلا ولا لا؟!  
ثم منغير الباسورد اذا تم التحقق, لكن ممكن الباسورد الجديدة تكون فيها مشاكل ما تتبع معايير ال Identity (تحتوي على حرف كابيتال حرف سمول ورقم ورقم مميز,..) منرجعله BadRequest   
غير هيك تم بنجاح  
  
ملاحظة : ChangePasswordAsync() من حالها بتحدث Password, PasswordHash وبتنادي داخليا على UpdateAsync(user) وبتم التحديث في DB تلقائيا**

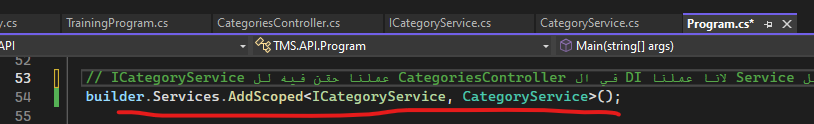
**CategoriesController**

****

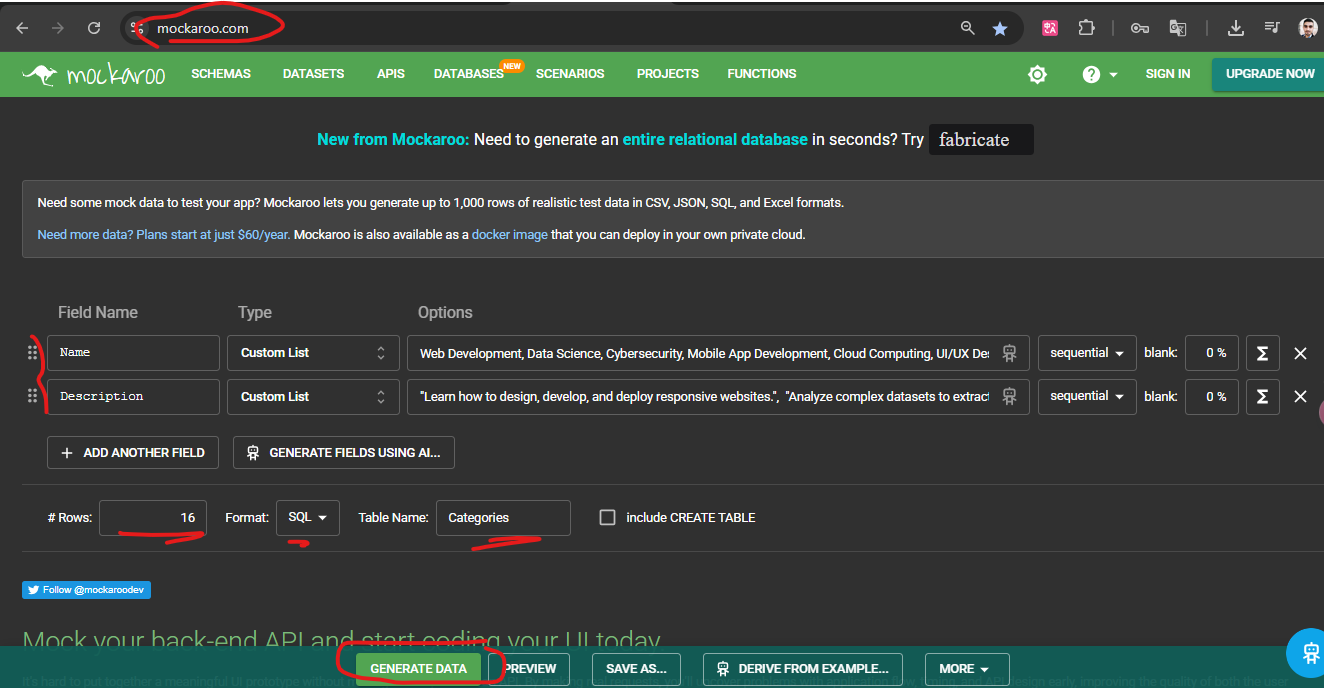
**في ال CategoriesController بعمل DI مع ال ICategoryService عشان اول ما اعمل run يفتح ال Scalar ويتصل مع ال ICategoryService ومن هناك ال CategoryService بتورث منها**

****

****

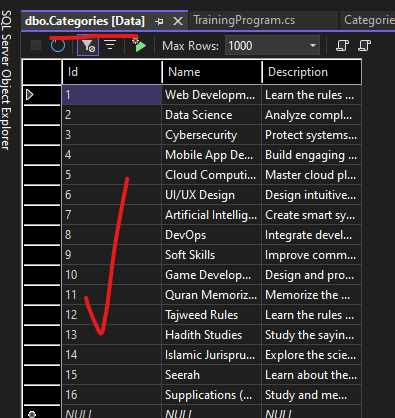
**ثم في ال CategoryService بتتصل مع ال DB ولا ننسى ان نعمل خدمة ال DI ال ICategoryService مع ال CategoryService لانو ال ICategoryService مش رح تقدر تتصل مع ال DB من حالها**

**هيك كل الأمور تمام وجاهز أي Action في ال CategoriesController يشتغل عادي.**

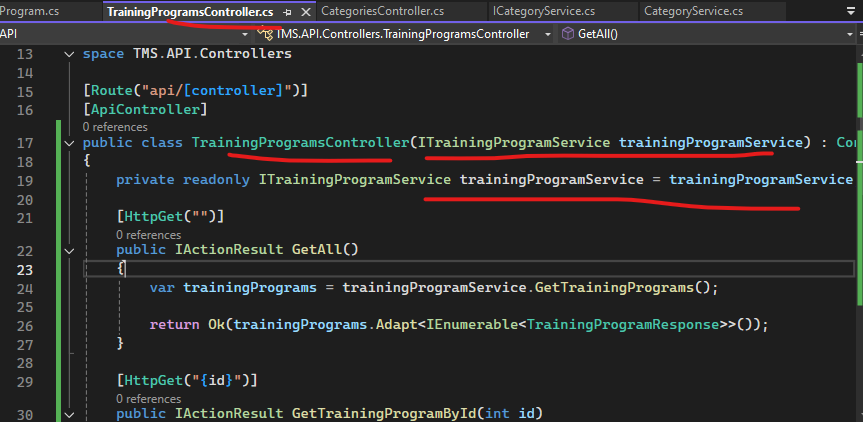
****

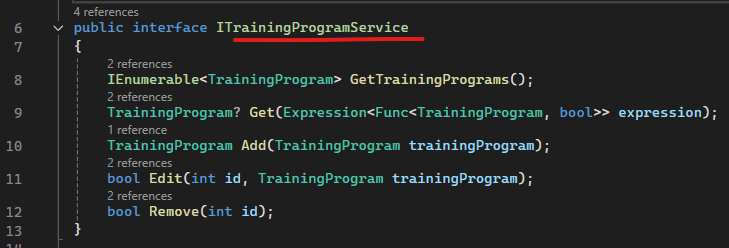
**الان دنضيف داتا لل Category**

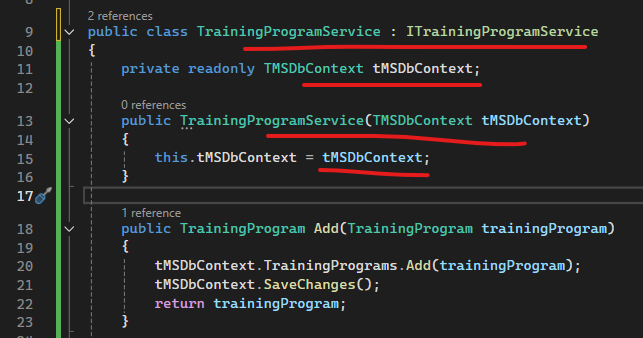
****

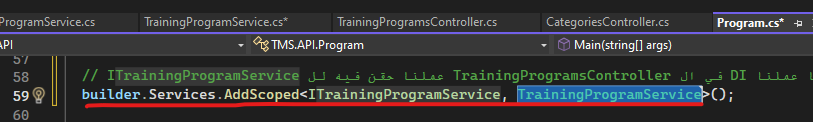
****

**كلو تمام.**

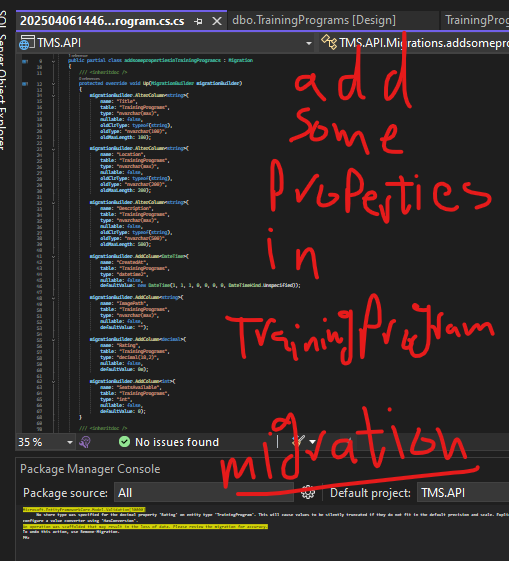
**TrainingProgramsController**

****

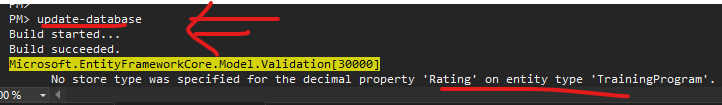
****

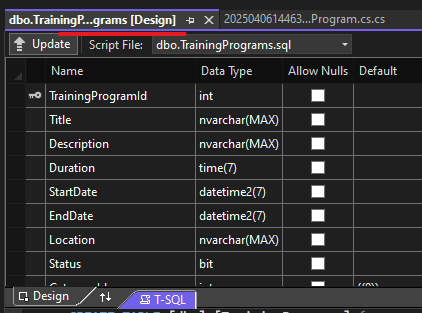
****

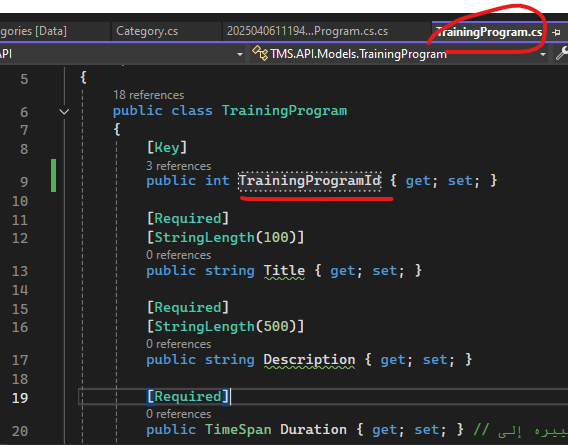
**هيك كلو شغال تمام.**

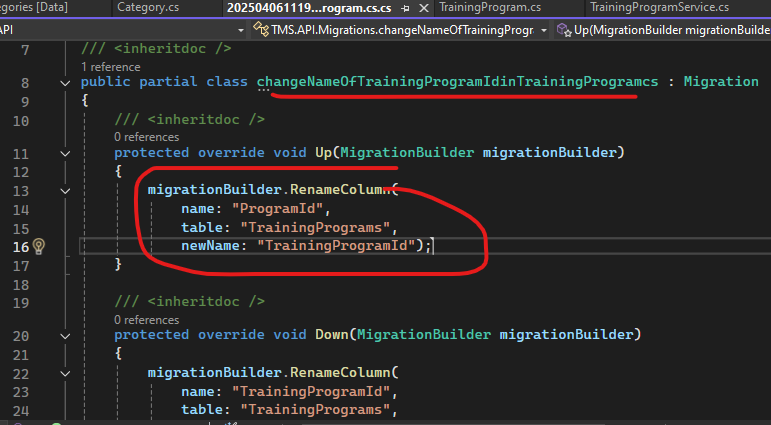
****

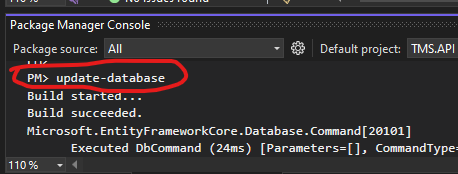
**ثم update-database**

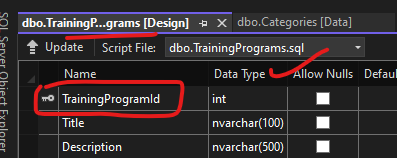
****

****

****

**  
انا هون غيرت اسم ال P.K كان بالبداية ProgramId الان خليته TrainingProgramId**

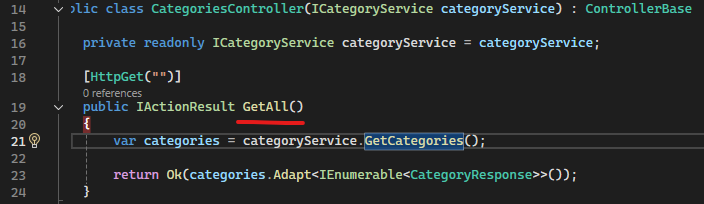
**نقوم بعمل ال migration ومن ثم update-database**

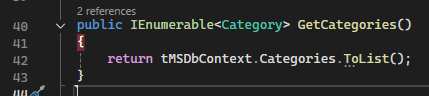
****

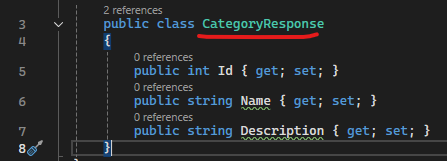
**كله تمام**

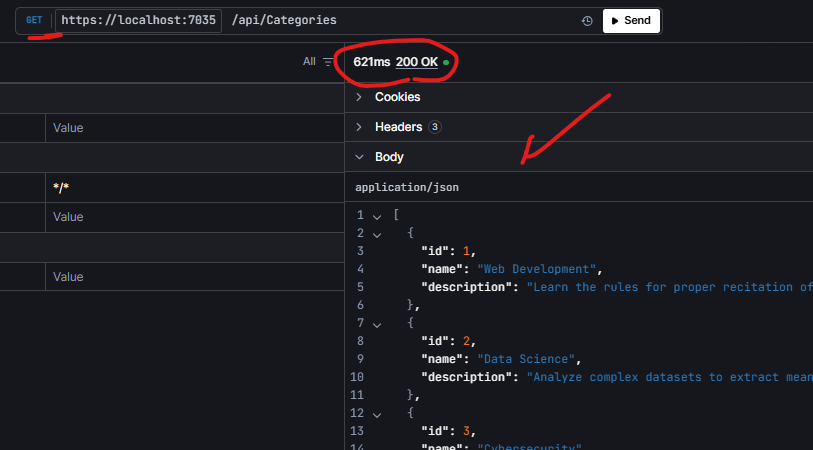
**CategoriesController Testing**

1. **GetAll()**

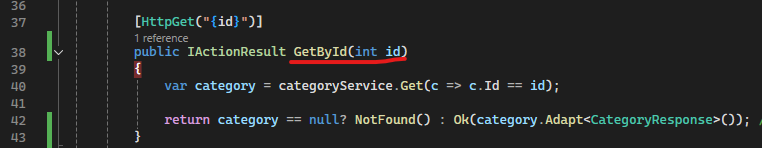
****

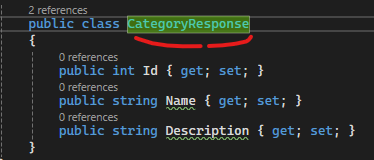
****

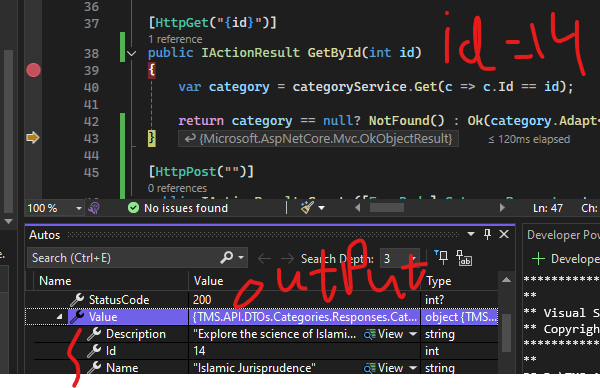
****

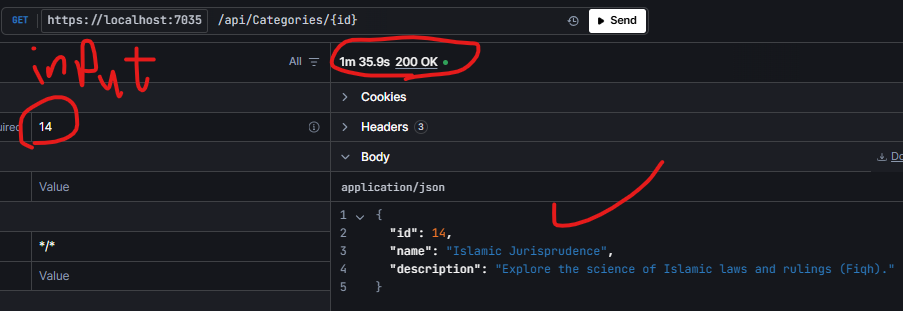
****

**شغال مية المية**

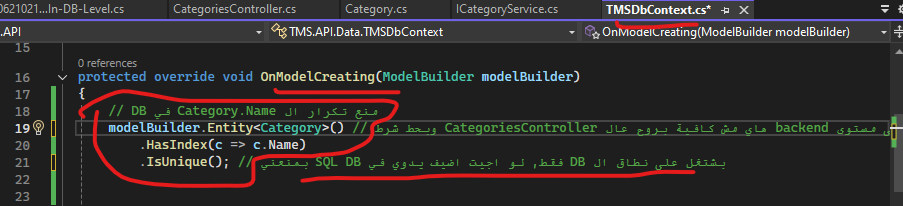
1. **GetById()**

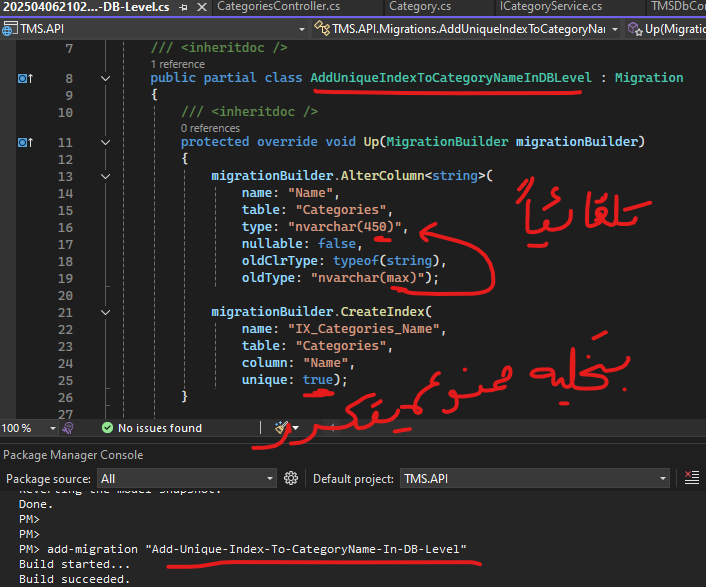
****

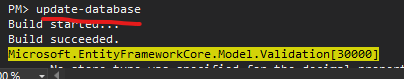
****

****

**تم بنجاح.**

****

****

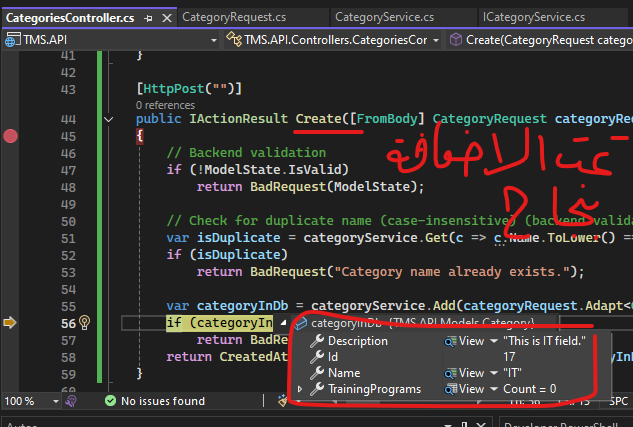
****

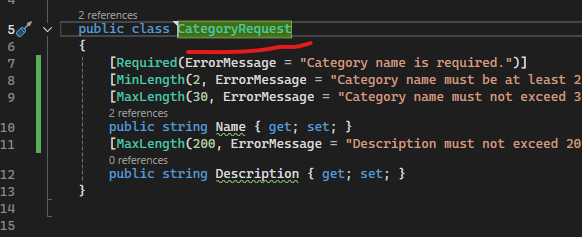
**عمل migration ثم update-database.**

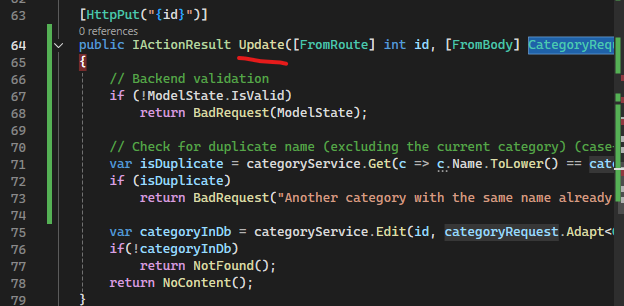
**هاي بس بتكون عمستوى ال DB لو انا بدي اضيف يدويا في SQL بمنعني, لكن عشان امنع من ال backend في ال scalar وتكون الأخطاء واضحة لانو اذا راح على ال DB ما رح يفهم الخطا بشكل مباشر لذلك بعمل backend validation لازم احط شروط في ال CategoriesConroller**

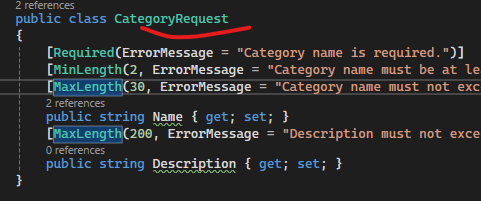
**بهاد الأسلوب بعمل backend validation كمان بحيث امنع لما ادخل بال Scalar كمان (على مستوى الفرونت لو بدو يضيف او يعدل بتجيه رسالة   
Category name already exists)**

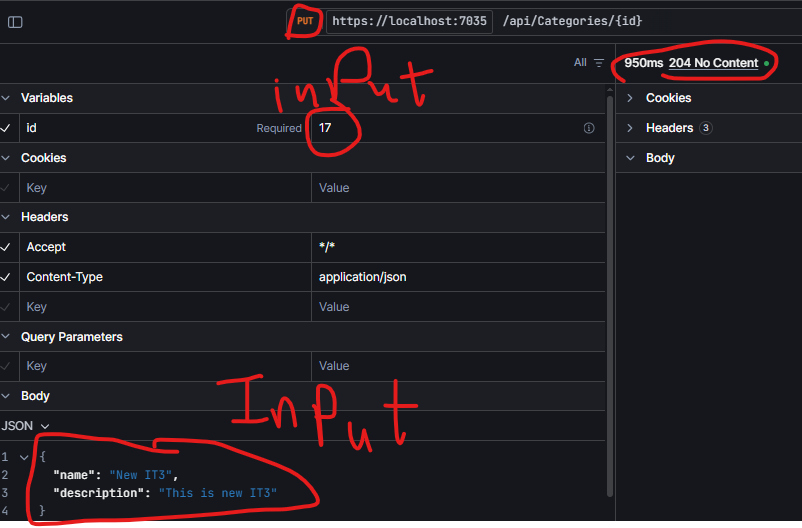
1. **Create()**

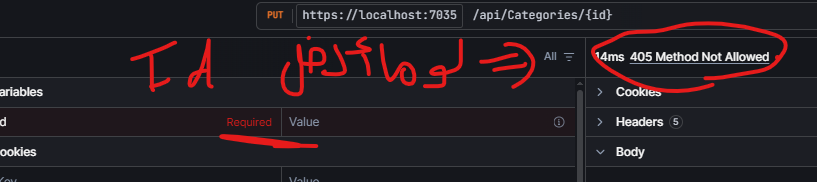
****

****

1. **Update()**

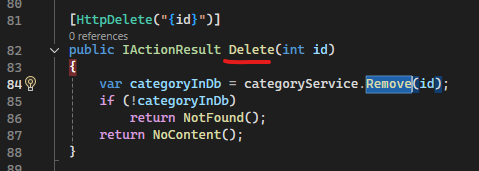
****

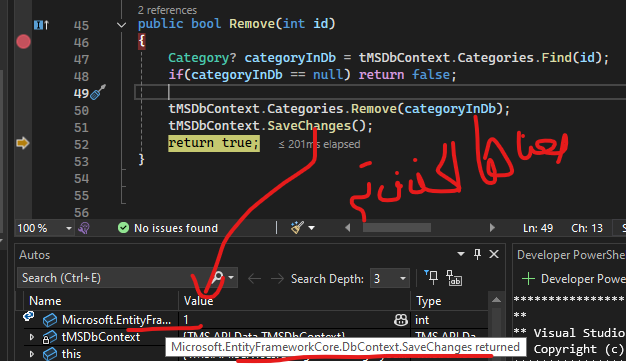
****

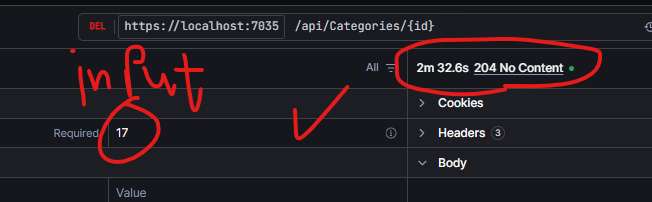
**شغال تمام**

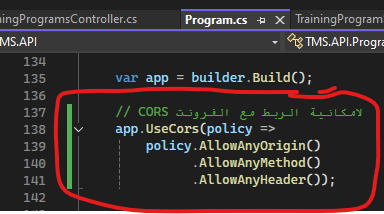
**ملاحظة :**

1. **Delete()**

****

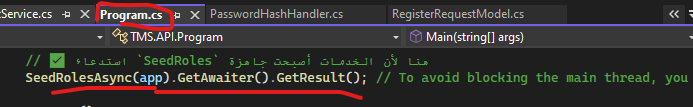
****

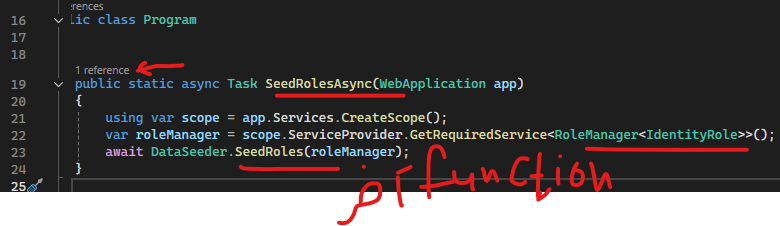
****

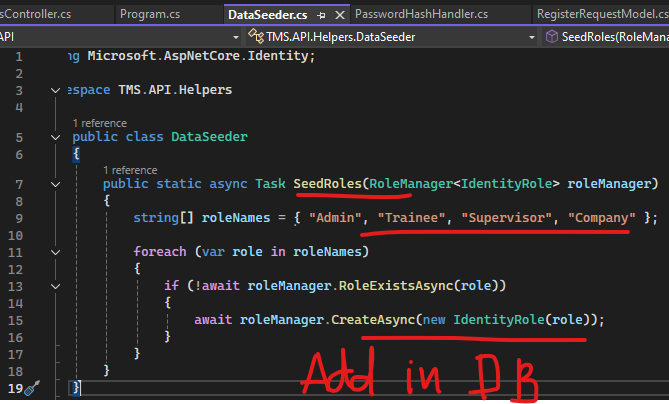
**تم الحذف بنجاح.**

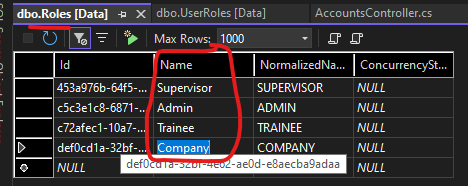
**للربط مع الفرونت ويشتغل  
ال API's صح.**

**TrainingProgramsController Test**

**توضيح كيف يتم بالبداية تهيئة انو عنا ال 4 أنواع من ال Actors فقط   
[Trainee/Admin/Company/Supervisor]**

**هون انا بستدعي ال function الي هو SeedRolesAsync()**

**  
يتم هنا تهيئة ال scope وعمل ال Service المطلوبة ثم يتم استدعاء function ال SeedRoles لنبدا بانشاء ال 4 أنواع فعليا.**

**هنا منضيف عال DB بحيث اذا شغلنا البرنامج وما كان احد هؤلاء ال 4 أنواع موجود في DB منضيفه على DB من خلال ال function ال CreateAsync()**

**هيك فعليا بكونوا موجودين بال DB  
مهيئين بالاصل.**