# API: Application Programming Interface

* ال api بمنتهى البساطة هي backend فقط لايحتوي على أي interface تتفاعل معه وحتى تتفاعل معه لازم يكون application اخر
* يحتوي على مجموعة endpoints تتعامل معها
* الميزة بال api (بغض النظر عن الtechnology المبينة فيها سواء dotnet, nodejs .. ) بتحسن تستخدمه بأي نوع من أنواع ال applications فبالتالي صار عندي backend واي front-end بيتعامل معه

# شرح أهم الملفات الرئيسية للمشروع:

* launchSetting.Json :
* ملف يحتوي على ال local IIS Configurations اللي عم تشتغل development عليه
* يحتوي على معلومات مثل:
* applicationUrl
* windowsAuthonetication: false
* anonymousAuthentication: true
* enviromentVariables:{

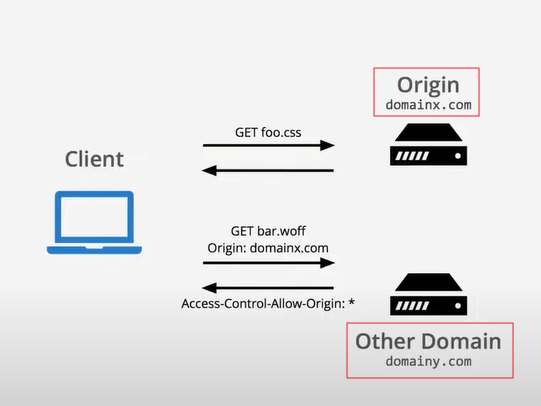
“ASPNETCORE\_ENVIROMENT”:”Development”

}

* بكون موجود فقط على ال localhost
* appsettings.json :
* يحتوي على معلومات مثل ال connectionString
* كمان فيك تحط your own values بال key and value تبعها
* Program.cs:
* يحتوي على ال builder تبع المشروع اللي انت رح تصير تضيف عليه ال services اللي رح تستخدمها

# CORS: Cross Origin Resource Sharing

* ال cors بالحالة الإفتراضية بتكون disable : يعني لو أنا عامل publish لل api تبعي ضمن network معينة فبالتي أي request بيجيني من جوا نفس ال network برد عليه ال api وفي حال اجاه ال request من خارج هي ال network بيرفضه ال api فبهي الحالة لازم أعمل enable لل cors ضمن ال api حتى يتعامل معي هي ال requests
* هي النقطة كتير مهمة جدا جدا بالنسبة لل security تبع ال api



* حتى أعمل enable لل cors شغلتها كتير بسيطة ببساطة أنا محتاج أضيف سطرين بال program.cs واحد بال service والتاني ضمن ال middlware
* بالنسبة للسطر تبع ال service مومهم الترتيب تبعه:

builder.Services.AddCors();

* بالنسبة للسطر التاني يفضل اضافته قبل ال UseAuthorization

app.UseCors(c=>c.AllowAnyHeader()

.AllowAnyMethod()

.WithOrigins("https://4200"));

app.UseAuthorization();

# Add Database Models

* رح نشتغل على MoviesApi
* كالعادة رح نزبط قصص ال ef وال dbcontext مارح اشرحهن لأنهن سهلين
* اضافة جدول Genres رح يحتوي على تصنيفات الأفلام (أكشن, كوميدي,..)
* ملاحظات مهمة:

1. أنا لما بدي عرف property على أنها required فيني أعمل هاد الشي من مكانين:

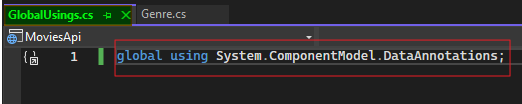
* data annotation
* fluent api

مع الdotnet6 وطلوع كل ال properties are Required وبتشوف هاد الشي من ال csproj:

<Nullable>enable</Nullable>

1. Global Usings:

* هي أنك تنقل ال usings تبع ال namespaces اللي انت بتستخدمها كتير ضمن global class بحيث أي كلاس عندك رح يشوف كل ال usings الموجودة ضمنه بدل ماتضل تكتب using بكل كلاس لحال
* ضمن المشروع ككل بضيف كلاس اسمه Global Usings وبضيف فيه كل ال usings اللي بدي ياها:



# Add Genres Controller

* مبدئيا رح ضيف ال ApplicationDbContext ضمن ال controller مباشرة وبعد ماننتهي تماما من ال endpoints رح نبلش نحكي عن ال services وننقل الأجزاء اللي عم تتعامل مع الداتابيز جوا هي ال services ورح نضيف كمان ال AutoMapper اللي رح يخلي الكود تبعنا منظم بشكل أكتر
* رح ضيف ال endpoints التالية:

1. public async Task<IActionResult> Get()

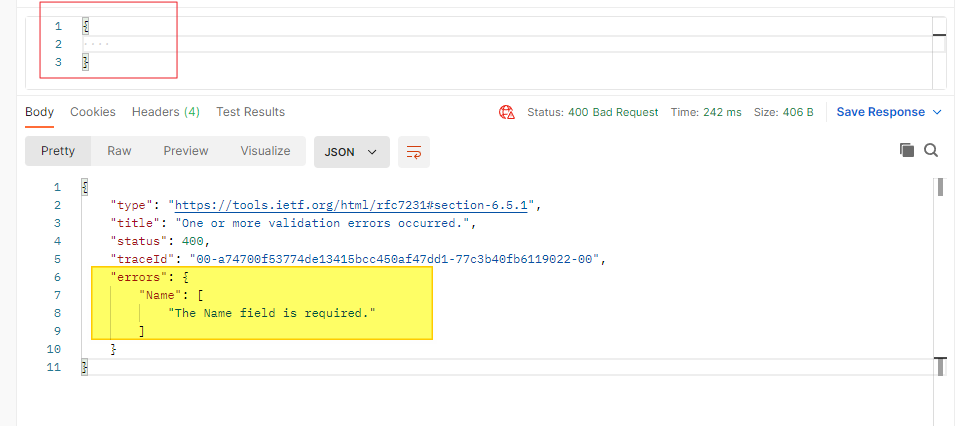
* ملاحظة: أي شي رح يرجع للفرونت وينعمله render داخل list يفضل تعملها ترتيب alphabet وماتتركه على ترتيب ال id

1. public async Task<IActionResult> Post(CreateGenreDto createGenreDto)

* بما انه ال id رح generated from database لاحظ هون اني لمابدي ضيف genre بعتت CreateGenreDto
* وظيفة ال dto Data Transfer Object هي نقل الداتا من وإلى ال api
* بعد ما أنشأ ال genre المفروض رجع ال genre اللي أنشأته بالداتابيز لأنه ممكن الشخص اللي شغال على front-end بكون مستني ال id اللي انشأت فيه هاد ال object أو ال created on
* في حال جربت ابعت ال genre body فاضي وهو مطلوب رح يرجع list بالأخطاء بدون مااحتاج اني اعمل manual valid متل العادة واتحق ق من حالة ال model ورح يعمل لحاله (auto validate)

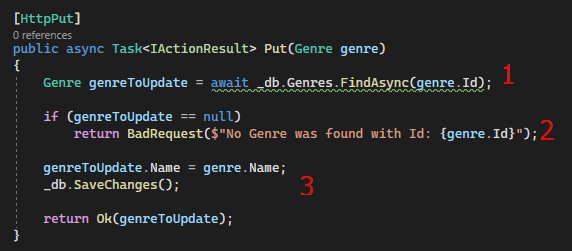
if (!ModelState.IsValid)

return BadRequest();



1. public async Task<IActionResult> Put(Genre genre)

في بعض القصص لازم راعيها عن ال update من ال 1 لل 3 كما موضح بالشكل:



1. public async Task<IActionResult> Delete(int id)

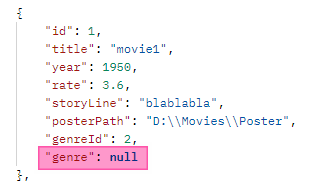
بس لازم انتبه على قصة التحقق من وجود ال genre متل ماعملت بال put بالزبط

# Add Movies Controller

* بعد ماخلصنا ال Endpoints تبع ال genres رح نبلش بال Moves endpoints
* رح أعمله كلاس Movie وضيفه على ال ApplicationDbContext
* رح ضيف ال endpoints التالية:

1. public async Task<IActionResult> Get()

* لاحظ أنه ال genre رح ترجع ب null لأني لما أعمل selection للداتا تبع ال movies هو ما بيعمل select لل related data الموجودة ضمن جدول ال genres



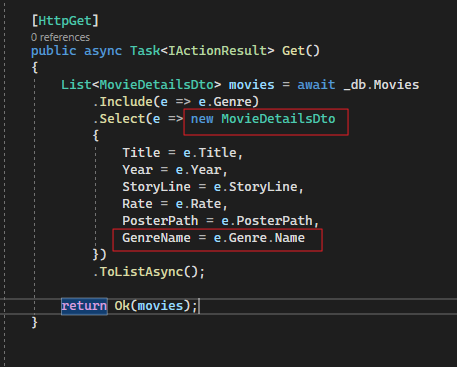
* وحلها بسيط جدا بإستخدام ال Include (بحددله شوهي ال navigation properties اللي أنا محتاج أعملها include معي وأنا عم return للداتا تبع ال movies)



* بس لاحظ أنه رجع الgenre على شكل Complex Object وهاد الشي بقابله الكويري التالية (Include(eager loading) is join):

|  |  |
| --- | --- |
|  |  |

* وطبعا اذا بدك ترجع بس الإسم بتعملها Dto :



|  |  |
| --- | --- |
|  |  |

1. public async Task<IActionResult> Get(int id)

* طبعا حتى جيب اسم ال genre ل movie **واحد** بدي **أعمل Include وهون رح يطلعلي error اللي هو اني مافيني استخدم find مع ال include وبستخدم بدالها ال SingleOrDefault**



1. public async Task<IActionResult> Post(CreateMovieDto createMovieDto)

* ملاحظة أنا رح ابعت صورة ال poster فبالتالي رح اتعامل مع صور وبشكل عام دائما لما تتعامل مع files في شغلتين اساسيتين لازم تدير بالك منهن وهنن ال حجم الملف (لازم تحدد ال file max size ) ونوعه (تحدد extensions معينة بتقبلها فقط)
* لازم أتأكد أنه ال |genreId المبعوت يكون موجود

if (!await \_db.Genres.AnyAsync(e => e.Id == createMovieDto.GenreId))

return BadRequest("Invalid Genre Id");

1. [HttpGet("getByGenreId")]

public async Task<IActionResult> GetByGenreId(int genreId)

* هي مشان تجبلي genre واحدة مع الأفلام تبعها
* لاحظ هون قصدا تركتلك [HttpGet("getByGenreId")] مشان ميزها عن غيرها وبالنسبة لل genreId بمرره ضمن ال Headers بال postman (key : value)

1. public async Task<IActionResult> Put(int id, [FromBody] CreateMovieDto movie)

* انتبه مع ال update لازم تحقق من نفس الشروط تبع ال post مثل genreId,posterSize,posterExtension …

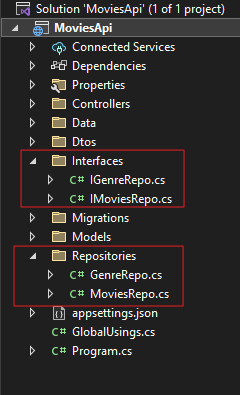
1. public async Task<IActionResult> Delete(int id)

# Code Refactoring

* بعد ماخلصنا كل ال endpoints اللي كنا محتاجينها بال apis تبعنا الان سوف نقوم بتحسين الكود تبعنا ونعمله refactoring
* سوف أقوم بتحسين الأمور التالية:

1. Add Genres Service:

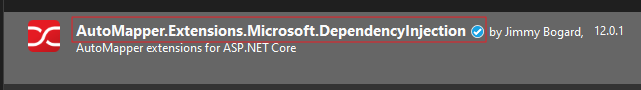
* كنت عم اعمل inject لل ApplicationDbContext مباشرة ضمن ال Controllerوهاد الشي اله كتير سلبيات والحل اني استخدم ال Repository Pattern



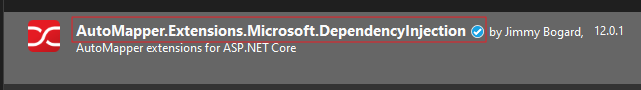
1. Use Automapper:

* أخر شي رح نعمله بهي السلسة اننه رح نستخدم ال automapper بحيث الكود تبعنا يكون شكله أفضل ومنظم بشكل أكتر(**بغير شكل الداتا من غير ما أعمل map كل property بال property اللي بتقابلها بال Dto**)
* حتى طبقه عندي بالمشروع:

1. لازم نزل الباكيج:



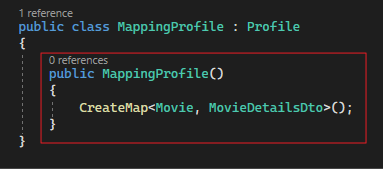
1. أعمله register بال program.cs:



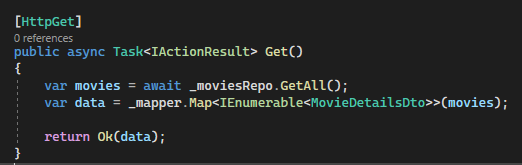
1. Add Mapping Profile

* بضيف مجلد Helper وبنشأ ضمن كلاس MappingProfile : Profile

1. جوا ال constructor بضيف ال mapping تبعي



1. اخر شي بعدله بال controller



هيك منكون خالصين الكورس والخطوة التالية هي انك تعمل Secure لل apis عن طريق ال JWT

Course Reference:

<https://www.youtube.com/watch?v=D_3Ycq1v4mI&list=PL62tSREI9C-c_yZ0a7Yui1U22Tv4mBjSF&ab_channel=DevCreed>

* ملاحظة مهمة **Update and Delete** هنن **Sync** حصرا