What is LINQ?

LINQ: Language Integrated Query

1. ليش منستخدمها؟

### - منعمل من خلالها كويري بتتعامل مع أي نوع من أنواع ال

Data sources: Oracle, SQL server, objects in memory, xml ...

- تخيل أنه عندي دوت نت ابليكشيشن والداتا بيز تبعه على سكوال فبالتالي الديفيلوبر لازم يتعلم ال

ADO. Net + SQL Server

وازا الداتا بيز كانت

XML -> XPath + XML Documents

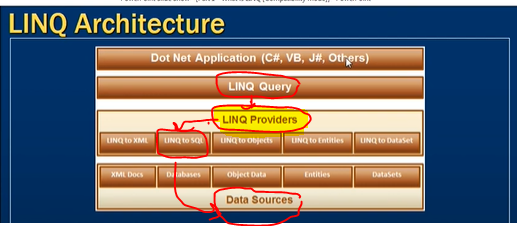
الLINQ حلت كل هي المشاكل حيث انها مكنتك من انك تتعامل مع اي داتا سورس وذلك بنفس الكود ستايل (**Similar Coding Style**)

يعني من الاخير بتكتب نفس الكود لكل أنواع الداتابيزات

1. Compile Time Error Checking + Intellisense
2. LINQ Architecture:

LINQ Providers: ببساطة مهمته تحويل الLINQ Query الى شكل بيفسهمه ال Data Source

في مثالنا هو بالتحديد **LINQ TO SQL** يقوم بتحويل ال LINQ query الى **Transaction Query** بيفهما ال DataSource



Writing LINQ Queries

1. عندي طريقتين لإكتب ال LINQ Query :

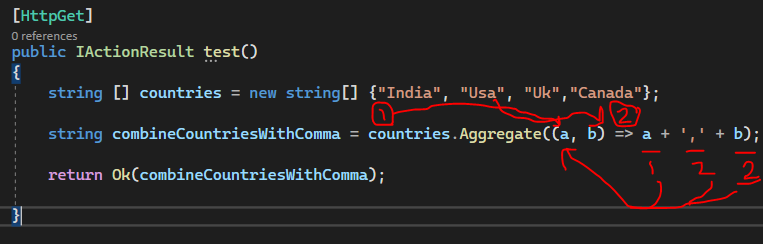
* Using Lambda Expression
* Using Query Expression

وماعندي أي فرق بال Performance بيناتهم (الأمر يعود لك بحرية الإختيار)

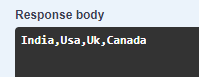
ملاحظة يتم تحويل ال Query Expression الى Lambda Expression قبل مايتم تنفيذها

Aggregate function in LINQ

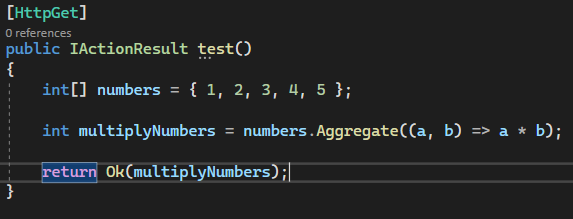
1. أريد جمع اسماء الدول مع فاصلة بينهم:



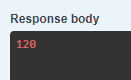
Output:



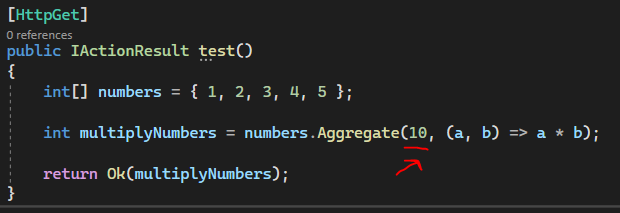
1. ضرب جميع الأرقام مع بعضها:



Output:



1. ضرب جميع الأرقام مع بعضها مع **تمرير قيمة بدائية**:

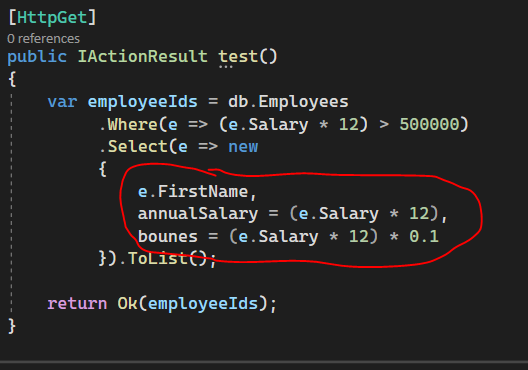


Output:



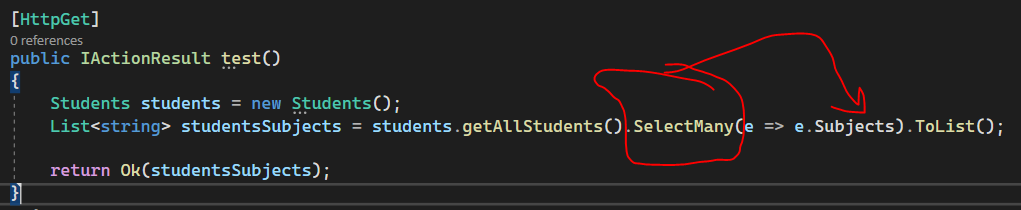
Projection Operators

ببساطة هي شلون بدي رجع النتيجة بال Select:

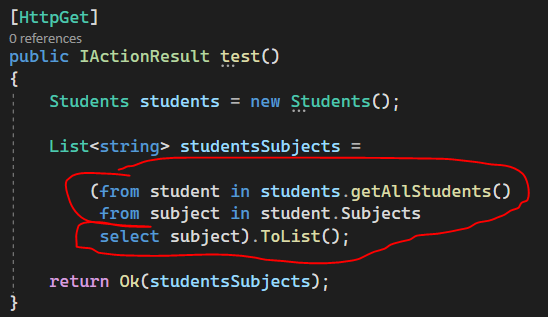


SelectMany Operator

لما بدي جيب list داخل List (يعني بدي جيب كل المواد لكل الطلاب)



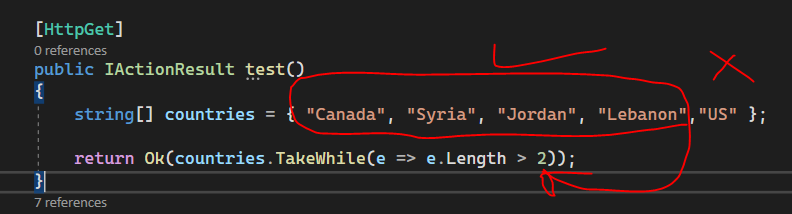
**طريقة أخرى للحل (لاحظ انه استخدمنا 2 nested from)**



Partitioning Operators

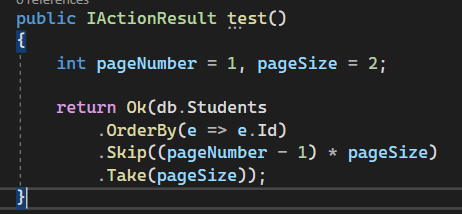


TakeWhile/ SkipWhileمن اسمها بترجع حسب شرط معين



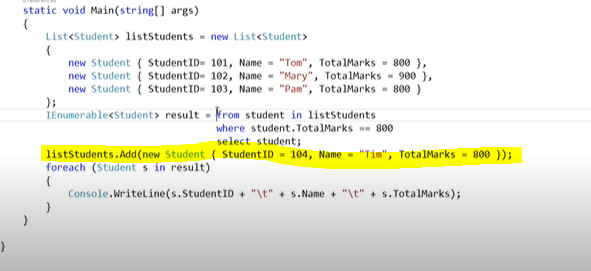
Common Use With:

**Implement paging using skip and take**



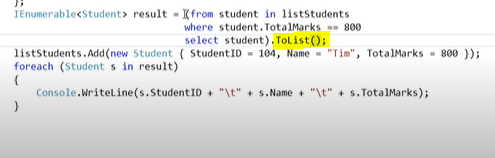
LINQ query deferred execution

التنفيذ المؤجل



**حسب الكويري اللي فوق فأنه يتم تأجيل تنفيذ الكويري لحين القيام بعملية الطباعة فبالتالي سيطبع معلومات TIM**

**وفي حال بدي خليه ينفذ مباشر بضفله ToList() وهيك مارح يطبع TIM**

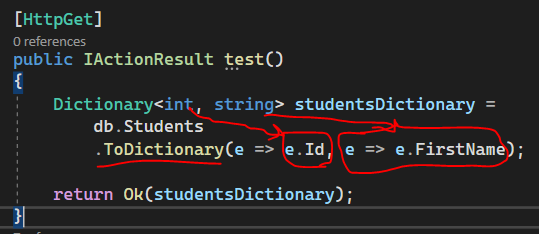


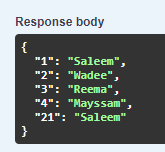
**LINQ Operators Categories based on execution behavior:**

1. Deferred/Lazy Operator: select, where, skip, take...
2. Immediate Operators: ToList(), Count(),Min(),Max(),…

Conversion Operators in LINQ

ببساطة ToList(), ToArray() ….

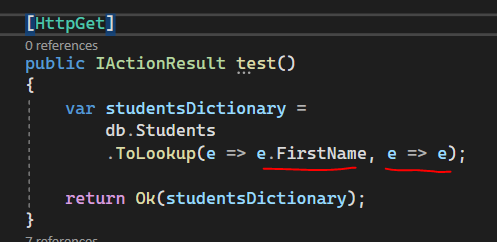
**ToDictionary (Key Value Pairs)**



ملاحظة مهمة : مع ال Dic يجب أن يكون ال Id Unique والا بيعطيك ايرور

وفي حال بدي جيب شي مكرر بستخدم ToLookup كما في المثال التالي( بدي جمع الطب حسب الاسم المكرر أوبدي جمع الموظفين حسب ال job Title تبعهن):

**ToLookup (Key Value Pairs)**

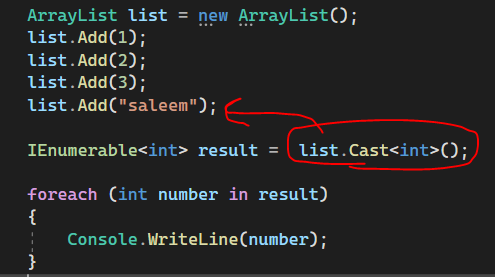




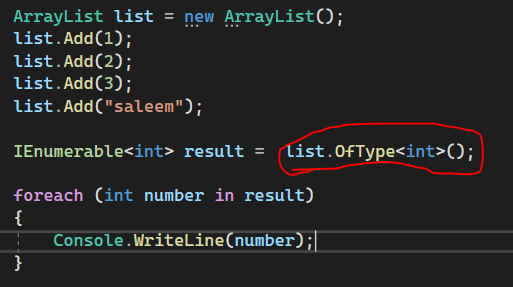
Cast and OfType operators

ببساطة ازا كان عندي List وفيها عناصر من انواع مختلفة وحابب اعملهن cast لنوع معين

هون رح يضرب:



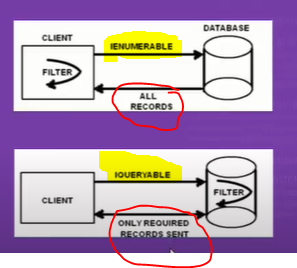
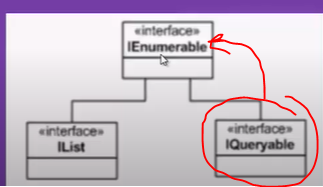
هون مارح يضرب (رح يتجاهل كل شي غير ال int):



Enumerable and AsQueryable

IEnumerable<Student>: بجيب الداتا كلها ك List وبعدين يتعامل معها من جوا التطبيق في الميموري

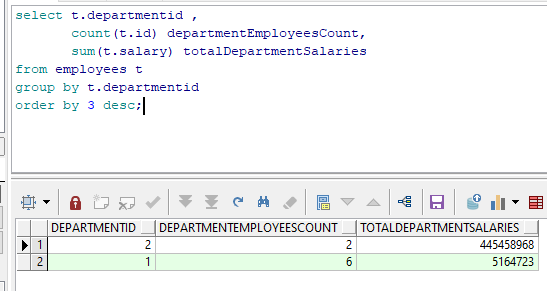
IQueryable<Student> : بجيب الداتا جاهزة من الداتا بيز (بعد الفلترة)

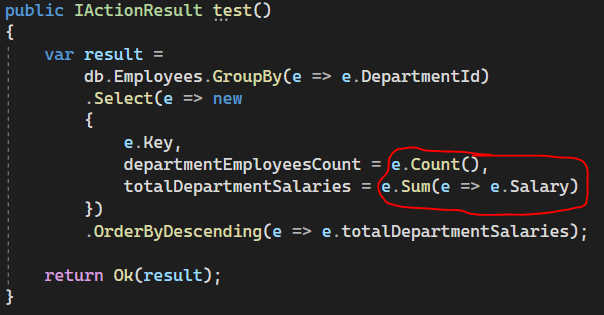
 

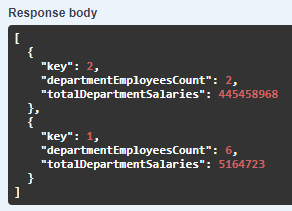
GroupBy

يجب مراجعة ال github الخاص بها فيه أمثلة حلوة وسهلة

معرفة عدد الموظفين في كل قسم مع معرفة مجموع الرواتب لكل قسم أيضا مرتبة تنازليا حسب الراتب:

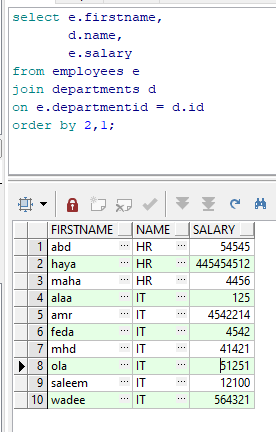


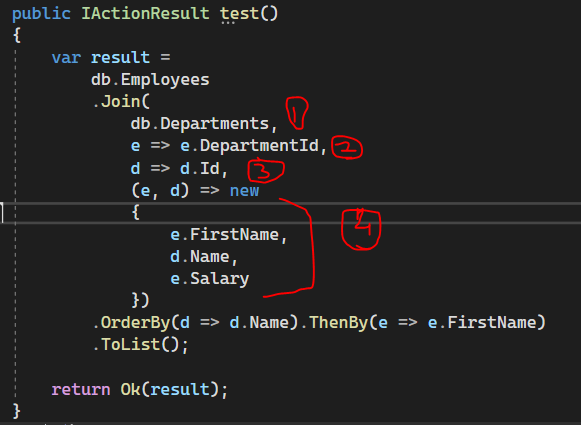




Join Tables

بدي جيب الموظفين مع اسماء الاقسام تبعهن





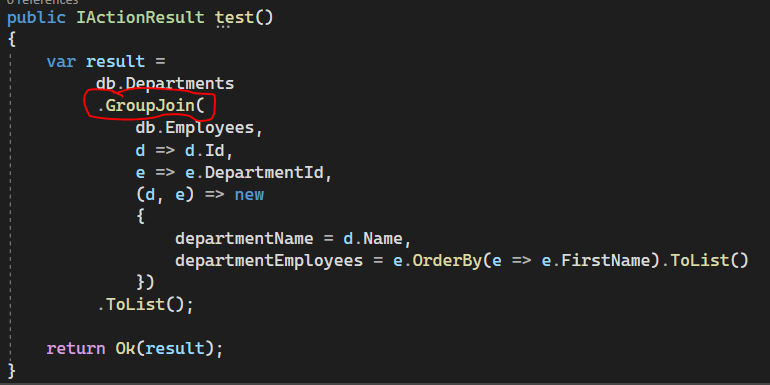
**معنى الأرقام بالصورة كالتالي:**

1. اسم الجدول اللي بدي join معه
2. اسم العمود من الجدول الاول
3. اسم العمود من الجدول الثاني
4. **Result Selector** : بجمع اسماء الاعمدة ضمن قوسين ومن ثم أختار الاعمدة من الجدولين اللي بدي ياها ترجع

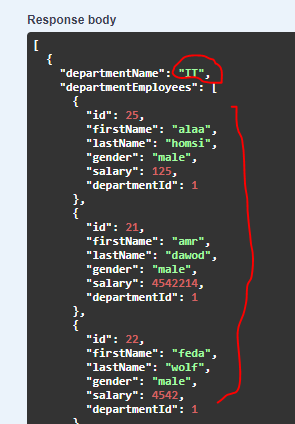


Group Join

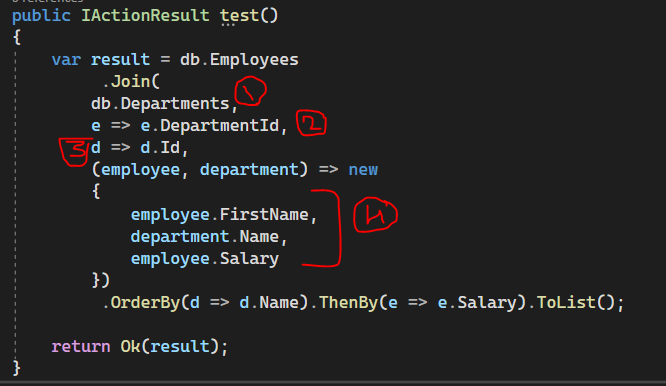
تحت كل اسم قسم بدي أسماء الموظفين تبعه

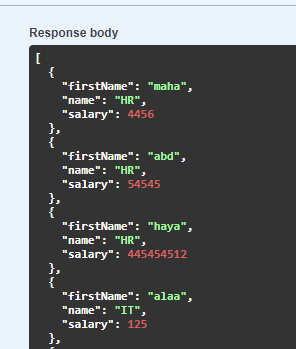


نفس ترتيب ال join العادية



Inner Join





|  |  |
| --- | --- |
| **Join** | **Group Join** |
| Flat Result | Hierarchical Result |
| جيب كل موظف مع قسمه مع تكرار القسم لكل موظف بنفس السطر | جيب القسم مع الموظفين اللي فيه مجمعين حسب اسم القسم |
| Like Inner Join In sql | Like Outer Join In Sql |