LINQ Quick Reference

.Net Framework with Web Databases
FALL SEMESTER 2019

LINQ Methods (requires using System.LINQ)

Highlighted methods are not deferred.

Filtering

Function	Description
Where(*x* => *clause*)	Returns the sequence of items where the *clause* is true.
First()	Returns only the first item. Throws an exception if there are no items.
FirstOrDefault()	Returns only the first item. Returns null if there are no items.
First(*x* => *clause*)	Returns only the first item where the *clause* is true. Throws an exception if there are no items where the *clause* is true.
FirstOrDefault(*x* => *clause*)	Returns only the first item where the *clause* is true. Returns null if there are no items where the *clause* is true.

Counting

Function	Description
Any()	Returns true if there are any items in the result set.
Any(*x* => *clause*)	Returns true if there are any items in the result set where the *clause* is true.
All(*x* => *clause*)	Returns true if the *clause* is true for all items in the result set.
Count()	Returns the number of items in the result set.
Count(*x* => *clause*)	Returns the number of items in the result set where the *clause* is true.

Aggregating

Function	Description
Sum(*x* => *column*)	Returns the sum for a particular *column.*
Average(*x* => *column*)	Returns the average value for a particular *column.*
Min(*x* => *column*)	Returns the minimum value for a particular *column.*
Max(*x* => *column*)	Returns the maximum value for a particular *column.*

Sorting

Function	Description
OrderBy(*x* => *column*)	Sorts the result set by a particular *column* in ascending order.
OrderByDescending(*x* => *column*)	Sorts the result set by a particular *column* in descending order.
ThenBy(*x* => *column*)	Used to sort the results by multiple columns.
ThenByDescending(*x* => *column*)	Used to sort the results by multiple columns.

Other Methods

Function	Description
Select(*x* => *result*)	Projects each item into a new form. Often used when you only want some of the columns in the result set.
Skip(*n*)	Discards the first *n* items from the result set and keeps the rest.
Take(*n*)	Gets the first *n* items from the result set and discards the rest.
Distinct()	Removes duplicates from the result set. WARNING: Distinct() is expensive in terms of execution time and memory, when there are few duplicates. If you can modify your query to remove duplicates some other way, that is often preferable. Duplicate results are often a sign that your joins are incorrect.
ToList()	Execute the query and store it in a list. WARNING: If you call ToList() too early, then you will be downloading a lot more data than you need, which will drastically hurt the performance of your application.

Async Methods (requires using System.Data.Entity)

All the non-deferred methods have async versions.

Note that all of these methods require an IQueryable<T>

- SaveChangesAsync()
- ToListAsync()
- FirstAsync(), FirstOrDefaultAsync(), etc
- AnyAsync(), AllAsync(), CountAsync(), etc
- SumAsync(), AverageAsync(), MinAsync(), MaxAsync()

https://docs.microsoft.com/en-us/dotnet/api/system.data.entity.queryableextensions?view=entity-framework-6.2.0