LINQ allows us to query a data source even in-memory collections to filter, shape, order, aggregate or locate elements.

We can query with LINQ (Language INtegrated Query) directly from a programming language.

What kind of data source can we query? Any with a LINQ provider.

LINQ to objects allows us to query any collection that implements IEnumerable<T>

LINQ to SQL works with SQL server databases.

LINQ to Entities works with entity framework

LINQ to XML works with XML documents.

There are many more and we can create them for custom data stores.

There are 2 ways to express a LINQ query:

* Query syntax in a form similar to SQL
* Method syntax. In .NET it’s implemented with a set of extension methods for the IEnumerable interface

Extension method – a method adding to an existing type without modifying the existing type.

Whats a delegate?

A type that represents a reference to a method. With a specific param list and return type.

Delegates are used to pass methods as arguments to other methods.

(Query Syntax)LINQ query returns an IEnumerable of the data type you’re querying for.

(Query Syntax)LINQ queries support deferred execution which means the LINQ query is not executed until the collection is iterated. This means when the query is defined, it is not executed. It is only executed when the result is iterated. For example .ToList();

(Method Syntax)The LINQ query method syntax is defined as extension methods for the IEnumerable interface. To use them we work with a collection that implements the IEnumerable interface. The methods take a delegate as an argument.

Func<one, two> - one is the arg type of the delegate. Two is the return type of the delegate.

=> defines a lambda expression

The method syntax will check each member of the collection. The problem is if we define methods to pass as args for the query methods then our code will become packed with many small methods. There must be a better way! Lambda expressions in place of the delegates. We pass lambda expressions for the query methods. This is considered better practice.

Where – defines filter

OrderBy – defines order criteria

Select – can return subsets of properties from elements.

GroupBy – defines grouping criteria.

Aggregation – can aggregate results within a collection. Example, Average.

First – find the first matching element. Also a Last method but the problem with both is they throw exceptions if not found. Better to use FirstOrDefault or LastOrDefault which will return a null or default value if not found.

Do:

Use LINQ. Saves iterating through ourselves.

Use Method syntax over query syntax

Define all query statements before iterating. It means only one iteration is required.

Avoid:

Using First or Last as if the result doesn’t exist will throw an exception. Use FirstOrDefault or LastOrDefault.

Iterate over the collection more than is required.

FAQ’s

1. What is LINQ?

Language Integrated Query. A way of querying collections from a .NET language.

1. What type of data source works with LINQ?

Whatever data source has a LINQ provider.

1. What is an extension method?

A method added to a class without editing the class.

1. What types do the LINQ extension methods extend?

IEnumerable

1. What is a delegate?

A type acting as a ref type to a method

1. What is a lambdaexpression?

A method defined on one line. A method that can be passed in place of a delegate.