LINQ

Language-Integrated Query

What Is LINQ?

LINQ (language-integrated query) bridges the gap between the world of objects and the world of data.

LINQ is used to interact with any class that implements the generic IEnumerable<T> interface. (For example: List<T>)

What Is LINQ?

LINQ uses two types of syntax

- Query Syntax
- Method Syntax

LINQ

All LINQ operations consist of three distinct actions:

1. Obtain the data source.
2. Create the query.
3. Execute the query.

Query Syntax

Looks Like SQL

1. Obtain the Data Source

In an Entity Framework Repository, it might look like this:

```
var allAuthors = context.Authors;
```

Otherwise it can be any object that inherits from IEnumerable, like this:

```
var allAuthors = new List<Authors> {author1, author2, author3};
```

2. Create a Query

Query your data source to get the data you need:

```
var allAuthors = context.Authors;

var authorsWithMultipleBooksQuery =
    from author in allAuthors
    where author.PublishedWorks.Count > 1
    select author;
```

LINQ Queries return an IEnumerable.

3. Execute the Query

Behind the scenes, your query doesn’t actually execute on the data store until you enumerate the query -- loop through it, perform actions on it, etc..

```
var allAuthors = context.Authors;  //Data source

var authorsWithMultipleBooksQuery = //Query creation
    from author in allAuthors
    where author.PublishedWorks.Count > 1
    select author;

var a = authorsWithMultipleBooksQuery.ToList(); //Query execution
```

LINQ Query Syntax Keywords

LINQ Query Keywords

MSDN Chart

LINQ Query Examples

```
var allAuthors = context.Authors;

var authorsNamedSteveQuery =
    from author in allAuthors
    where author.FirstName == “Steve”
    select author;

var authorsYoungestToOldestQuery =
    from author in allAuthors
    orderby author.BirthDate descending
    select author;

var authorsByLastInitialQuery =
    from author in allAuthors
    group author by author.LastName[0] //returns the 1st char
    select author; //query returns IEnumerable<IGrouping<Author>>
```

LINQ Query Examples, cont.

```
var allAuthors = context.Authors;
var allBooks = context.Books;

var booksByAuthorLastNameQuery =
    from author in allAuthors
    join book in allBooks on author.Id equals book.AuthorId
    group book by Author.LastName ascending
    select book; //Selects all the books and groups them by
                 //author's last name

var booksBySteveQuery =
    from book in allBooks
    join author in allAuthors on book.AuthorId equals author.Id
    where author.FirstName == "Steve"
    select book; //returns books where Steve is the author
```

Method Syntax

Lambda Expressions

Lambda Expressions

A lambda expression is an anonymous function that you can use to create delegates or expression tree types.

LINQ Method Syntax using lambda expressions is a functional programming style of interacting with a data source.

LINQ Method Syntax

Like query syntax, method syntax has three parts:

1. Data source
2. Query creation
3. Query execution

LINQ Method Syntax Examples

```
var allAuthors = context.Authors;

var authorsNamedSteve =
    allAuthors.Where(author => author.FirstName == “Steve”);

var authorsOldestToYoungestQuery =
    allAuthors.OrderBy(author => author.BirthDate);
    //.OrderByDescending would make this list youngest to oldest

var authorsByLastInitialQuery =
    allAuthors.GroupBy(author => author.LastName[0]);

var authorWithMyNameOrNullQuery = allAuthors
    .SingleOrDefault(author =>
        author.FirstName == “Kate”
        && author.LastName == “Williams”
    ); //If no authors meet the conditions, the method returns null
```

LINQ Method Syntax Examples, cont.

```
var allBooks = context.Books;

var booksGroupedByAuthorLastNameQuery =
    allBooks.GroupBy(book => book.Author.LastName);

var booksBySteveQuery =
    allBooks.Where(book => book.Author.FirstName == “Steve”);

var publishDatesOfStevesBooksQuery =
    allBooks.Where(book => book.Author.FirstName == “Steve”)
        .Select(bookBySteve => bookBySteve.PublishDate);
    //returns an IEnumerable<DateTime>

var publishDatesOfStevesBooksThisYearQuery =
    allBooks.Where(book => book.Author.FirstName == “Steve”)
        .Select(bookBySteve => bookBySteve.PublishDate)
        .Where(bbspd => bbspd.Year == DateTime.Now.Year);
```

Common Lambda Expression Methods

MSDN

- **Where** - returns the items that match a boolean condition
- **Select** - transforms an object into a new form
- **Single/SingleOrDefault** - returns the only item that matches
- **First/FirstOrDefault** - returns the first items that matches
- **Any** - returns true if any item meets boolean condition
- **All** - returns true if all items meet boolean condition
- **Min/Max** - transforms on object into a new form and return minimum or maximum of the transformed value.
- **OrderBy** - orders list by a specified key

THE END

Just Kidding

Here’s A Practice Assignment