

Easily Add PDF Word & Excel Function to Your .NET Apps

HTML clipboard

LINQ is a cool feature in C# 3.0. Most of the developers are struggling for the syntax and examples. Here I have collected various examples for each operator in LINQ and the equivalent Lambda Expressions.

Where

```
01. IEnumerable<Product> x = products.Where(p => p.UnitPrice >= 10);
02.
03. IEnumerable<Product> x =
04. from p in products
    where p.UnitPrice >= 10
06. select p;
```

Select

```
IEnumerable<string> productNames = products.Select(p => p.Name);
02.
      IEnumerable<string> productNames = from p in products select p.Name;
03.
04.
      var namesAndPrices =
05.
      products.
06.
      Where (p => p.UnitPrice >= 10).
07.
     Select(p => new { p.Name, p.UnitPrice }).
08.
      ToList();
09.
     IEnumerable<int> indices =
10.
      products.
11.
      Select((product, index) => new { product, index }).
12.
     Where (x \Rightarrow x.product.UnitPrice >= 10).
13.
     Select(x => x.index);
```

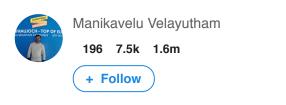
SelectMany

12.

customers.

```
01.
     IEnumerable<Order> orders =
02.
     customers.
03.
     Where(c => c.Country == "Denmark").
04.
     SelectMany(c => c.Orders);
05.
     var namesAndOrderIDs =
06.
     customers.
07.
     Where(c => c.Country == "Denmark").
08.
     SelectMany(c => c.Orders).
     Where (o => o.OrderDate.Year == 2005).
09.
10.
     Select(o => new { o.Customer.Name, o.OrderID });
     var namesAndOrderIDs =
11.
```

Don't miss Manikavelu Velayutham's next article



Take

01. IEnumerable<Product> MostExpensive10 =products.OrderByDescending(p => p.UnitPrice).Take(10);

Skip

01. IEnumerable<Product> AllButMostExpensive10 = products.OrderByDescending(p => p.UnitPrice).

TakeWhile SkipWhile

```
01. s.TakeWhile(p) s.SkipWhile(p)
```

Join

```
01.
     var custOrders = customers.
     Join(orders, c => c.CustomerID, o => o.CustomerID, (c, o) => new {
02.
03.
          c.Name, o.OrderDate, o.Total
04.
     });
05.
     var custOrders = from c in customers
06.
     join o in orders on c.CustomerID equals o.CustomerID
07.
     select new {
08.
          c.Name, o.OrderDate, o.Total
09.
     };
```

GroupJoin

```
01.
      var custTotalOrders = customers.
02.
      GroupJoin(orders, c => c.CustomerID, o => o.CustomerID, (c, co) => new {
03.
          c.Name, TotalOrders = co.Sum(o => o.Total)
04.
      });
05.
      var custTotalOrders = from c in customers
06.
      join o in orders on c.CustomerID equals o.CustomerID into co
07.
      select new {
08.
          c.Name, TotalOrders = co.Sum(o => o.Total)
09.
      };
10.
      var custTotalOrders = from c in customers
                                                         Don't miss Manikavelu Velayutham's next
11.
      join o in orders on c.CustomerID equals o.Custor
                                                         article
12.
      select new {
13.
          c.Name, o.OrderDate, o.Total
14.
                                                                 Manikavelu Velayutham
15.
      var custTotalOrders = from c in customers
16.
      join o in orders on c.CustomerID equals o.Custom
                                                                   196 7.5k 1.6m
17.
      from o in co
18.
      select new {
                                                                   + Follow
19.
          c.Name, o.OrderDate, o.Total
20.
21.
      var custTotalOrders = from c in customers
22.
      join o in orders on c.CustomerID equals o.CustomerID into co
23.
      from o in co.DefaultIfEmpty(emptyOrder)
```

select new {



ASK A QUESTION

```
01.
     IEnumerable < string > locations = customers.Select(c => c.City).
02.
      Concat(customers.Select(c => c.Region)).
03.
      Concat(customers.Select(c => c.Country)).
04.
      Distinct();
05.
      IEnumerable < string > locations = new [] {
06.
          customers.Select(c => c.City),
07.
              customers.Select(c => c.Region),
08.
              customers.Select(c => c.Country),
09.
      } .
      SelectMany(s \Rightarrow s).
10.
11.
     Distinct();
```

OrderBy / ThenBy

```
01.
      IEnumerable<Product> orderedProducts1 =
02.
      products.
03.
      OrderBy(p => p.Category).
04.
      ThenByDescending(p => p.UnitPrice).
05.
      ThenBy(p \Rightarrow p.Name);
06.
      IEnumerable<Product> orderedProducts1 =
07.
      from p in products
08.
      orderby p.Category, p.UnitPrice descending, p.Name
09.
      select p;
10.
      IEnumerable<Product> orderedProducts2 =
11.
      products.
12.
      Where (p => p.Category == "Beverages").
13.
      OrderBy(p => p.Name, StringComparer.CurrentCultureIgnoreCase);
14.
      IEnumerable<string> orderedProductNames =
15.
      products.
16.
      Where (p => p.Category == "Beverages").
      Select(p => p.Name).
17.
18.
     OrderBy(x \Rightarrow x);
```

GroupBy

```
IGrouping<string, Product>> productsByCategory =products.GroupBy(p => p.Category);
IGrouping<string, string>> productNamesByCategory =products.GroupBy(p => p.Category, p => p.Name);
```

Distinct

01. IEnumerable<string> productCategories =products.Select(p => p.Category).Distinct();

AsEnumerable

```
01. Table<Customer> custTable = GetCustomersTable()
02. var query = custTable.AsEnumerable().Where(c =>
```

ToArray

```
01. string[] customerCountries =customers.Select(c
```

ToList

```
01. List<Customer> customersWithOrdersIn2005 =customers.
02. Where(c => c.Orders.Any(o => o.OrderDate.Year == 2005)).
03. ToList();
```

Don't miss Manikavelu Velayutham's next article



Manikavelu Velayutham

196 7.5k 1.6m

+ Follow

```
Dictionary<int, Order> orders =customers.
     SelectMary(c => c.Orders).
                                                    ASK A OUESTION
     Where(o => o.OrderDate.Year == 2005).
04.
     ToDictionary(o => o.OrderID);
05.
     Dictionary<string,decimal> categoryMaxPrice =products.
06.
     GroupBy(p => p.Category).
07.
     ToDictionary(g => g.Key, g => g.Group.Max(p => p.UnitPrice));
```

ToLookup

```
01. Lookup<string,Product> productsByCategory =products.ToLookup(p => p.Category);
02. IEnumerable < Product > beverages = products By Category ["Beverage"];
```

OfType

```
01. List<Person> persons = GetListOfPersons();
     IEnumerable<Employee> employees = persons.OfType<Employee>();
```

Cast

```
ArrayList objects = GetOrders();
01.
02.
      IEnumerable<Order> ordersIn2005 =
03.
      objects.
04.
      Cast<Order>().
05.
      Where (o => o.OrderDate.Year == 2005);
06.
     ArrayList objects = GetOrders();
07.
      IEnumerable<Order> ordersIn2005 =
08.
     from Order o in objects
09.
     where o.OrderDate.Year == 2005
10.
     select o;
```

First

```
01. string phone = "206-555-1212";
    Customer c = customers.First(c => c.Phone == phone);
```

Single

```
01. int id=12345;
     Customer c = customers.Single(c => c.CustomerID == id);
```

ElementAt

```
01. Product thirdMostExpensive = products.OrderByDescending(p => p.UnitPrice).ElementAt(2);
```

Range

```
01. int[] squares = Enumerable.Range(0, 100).Select
```

Repeat

```
01. long[] x = Enumerable.Repeat(-1L, 256).ToArray()
```

Don't miss Manikavelu Velayutham's next article



Manikavelu Velayutham

196 7.5k 1.6m

+ Follow

Empty



ASK A OUESTION

```
bool b = products.Any(p => p.UnitPrice >= 100 && p.UnitsinStock == 0);
```

AII

```
01. IEnumerable<string> fullyStockedCategories =products.
02. GroupBy(p => p.Category).
03. Where(g => g.Group.All(p => p.UnitsInStock > 0)).
04. Select(g => g.Key);
```

Count

```
01. int count = customers.Count(c => c.City == "London");
```

Sum

```
01.
     int year = 2005;
02.
     var namesAndTotals = customers.
03.
     Select(c => new {
04.
         c.Name,
05.
              TotalOrders = c.Orders.
06.
          Where (o => o.OrderDate.Year == year).
07.
          Sum(o => o.Total)
08.
     });
```

Min

Max

```
01. decimal largestOrder = customers.
02. SelectMany(c => c.Orders).
03. Where(o => o.OrderDate.Year == 2005).
04. Max(o => o.Total);
```

Page

RENT FREND! What the heck is Bitcoin

Aggregate

```
01.  var longestNamesByCategory = products.
02.  GroupBy(p => p.Category).
03.  Select(g => new {
    Category = g.Key,
    LongestName = g.Group.
06.  Select(p => p.Name).
07.  Aggregate((s, t) => t.Length > s.Length ? t : s)
```

Don't miss Manikavel CONTRIBUTE article



Manikavelu Velayutham

196 7.5k 1.6m

+ Follow



LINQ

LINQ Oper

ASK A OUESTION



Manikavelu Velayutham 10P 500

I have over 11 years of IT industry experience with Microsoft technologies. I hold Masters degree in Computer Science and Applications. I am responsible for providing solution in Service Oriented Architecture and involvi... Read more

http://www.c-sharpcorner.com/members/manikavelu-velayutham

196 1.6m 1

4 9



Type your comment here and press Enter Key (Minimum 18 characters)



Nice article, Plz share some basic step by step query writing in linq for CRUD operation. Thanks in advance.

Raghav P Sep 01, 2017

1502 1 0 0 Reply

Very Truc

Very nice for article of Manikavelu Velayutham. Hope the features, me try more articles...!

Truong Minh Tuan Jul 26, 2017

1502 1 0 0 Reply



good one thannks

Sonu Chaudhary Jun 07, 2016

590 1.6k 101.1k 0 0 Reply



Very Nice...

Mukesh Kumar Tiwari Dec 10, 2014

872 764 111.2k 0 0 Reply



great work..really helpful for the new developers.Thanks

Satya

1502 1 0



Very good post but not for beginners.

Santosh Yadav

1255 249 68.2k



Very Useful and Handy.....

Siva Kumar ss

1499 4 1.8k

Don't miss Manikavelu Velayutham's next article



Manikavelu Velayutham

196 7.5k 1.6m

+ Follow

0 0 -

Nice and practical approach:)



ASK A QUESTION

Oct 25, 2011 0 Reply

Comment Using

| 0 Comments | Sort by Oldest |
|------------|----------------|
|------------|----------------|



Add a comment...

Facebook Comments Plugin



File APIs for .NET

Aspose are the market leader of .NET APIs for file business formats – natively work with DOCX, XLSX, PPT, PDF, MSG, MPP, images formats and many more!

TRENDING UP

- 01 Getting Started With Angular 5 And ASP.NET Core
- 02 Angular 5 App With ASP.NET Core 2.0 Web API
- 03 Server Side Custom Paging In Angular-UI Grid Using Web
- 04 Angular 5, ASP.NET Core CRUD For Inventory Management Using EF And WED AFT
- 05 Top 10 Web Application Security Risks In 2017

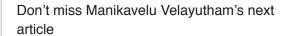




ASK A QUESTION

- 08 What Is SQL Operations Studio
- 09 Audit Made Easy Without Audit Log Part One
- 10 Angular 5 Basic Demo Project Overview

View All





Manikavelu Velayutham

196 7.5k 1.6m

+ Follow

Delh



d millions of develope ASK A QUESTION

Enter your email address

Sign Up

Learn ASP.NET MVC Learn ASP.NET Core Learn Python Learn JavaScript Learn Xamarin Learn Oracle More...

Home Events Consultants Jobs Career Advice Stories Partners

About Us Contact Us Privacy Policy Terms Media Kit Sitemap Report a Bug FAQ ©2017 C# Corner. All contents are copyright of their authors.

Don't miss Manikavelu Velayutham's next article



Manikavelu Velayutham

196 7.5k 1.6m

+ Follow