

ANDREY ZAVADSKIY – SQL AND .NET DEVELOPER, TRAINER, SPEAKER

Give me an URL to rely on, and I will shift the world

FEBRUARY 11, 2016 BY AZ

C#. Passing Lambda Expression As Parameter



Using delegates you can pass a lambda expression as a parameter to a function, and then use it in LINQ queries. This can be done with `Func<...>` delegates.

If you want to pass a lambda expression to be used, for example, in `Where` clause, you need a `Func<T, bool>` delegate. Just hover the mouse over `Where` in your code, and IntelliSense would show you the required syntax.

Here is a simple demo program:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4
5 namespace ConsoleApplication1
6 {
7     class Program
8     {
9         class Item
10        {
11            public int Id { get; set; }
12            public string Name { get; set; }
13        }
14
15        static void Main(string[] args)
16        {
17            List<Item> items = new List<Item>
18            {
19                new Item(){ Id = 1, Name = "Adam" },
20                new Item(){ Id = 2, Name = "Billy" },
21                new Item(){ Id = 3, Name = "Cecil" }
22            };
23
24            Console.WriteLine("Length == 5");
25            PrintItems(items, x => x.Name.Length == 5);
26        }
27    }
28 }
```

```
27     Console.WriteLine("Id is odd number");
28     PrintItems(items, r => r.Id % 2 == 1);
29 }
30
31 static void PrintItems(List<Item> list, Func<Item, bool> condition)
32 {
33     var selectedItems = list.Where(condition);
34
35     foreach (var item in selectedItems)
36     {
37         Console.WriteLine("{0}, {1}", item.Id, item.Name);
38     }
39 }
40 }
41 }
```

 **C#**