Linq: Language Integrated Query
Linq is a API

Advantage:

- Common to all datasource (sql,oracle,access,….)
  (i.e no need to change linq (query) if you want to change sql to oracle)
- Compile time query check in linq.

Header file:

- Using System.Linq;
- Using System.Xml.Linq;(xml to linq)

Controller File:(HomeController.cs)
==========

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Xml.Linq;
using EFwork.Models;

namespace EFwork.Controllers
{
    public class HomeController : Controller
    {
        //
        // GET: /Home/


        public ActionResult Index()
        {
            var db = new CustomerDataContext();

            var tb = db.Customers;

            IQueryable<Customer> q1 = from s in db.Customers
                                      //where s.ID > 5
                                      //where s.ID>3 && s.ID<=5
                                      //where s.Gender=="female"
                                      //orderby s.Name ascending
                                      select s;



            //var q2 = from s in db.Customers
            //                        select s;

            //var q3 = from s in db.Customers
             //                       select s.Name;
            //var q4 = from s in db.Customers
            //       select new { s.ID,s.Name,s.Gender,s.Email,s.Address };
```

```
            return View(q1);    // tb,q1,q2 accept,   q3 or a4  not accept because
filtered
        }

        // using for each

        public ActionResult ShowFilterData()
        {
            String names = "";
            var db = new CustomerDataContext();
            var tb = db.Customers;
            IQueryable<Customer> q1 = from s in tb
                                      select s;
            foreach (var cust in q1)
            {
                names = names + "   " + cust.Name;
            }
            ViewBag.c = names;
            return View();
        }
        // using aggregate function  count,max,min,sum (immediate execution)
        public ActionResult ShowCount()
        {
            var db = new CustomerDataContext();
            var tb = db.Customers;

            int cusCount = (from s in tb
                            select s).Count();
            ViewBag.count = cusCount;
            return View();
        }

        // immediate query execution ToList<source> (retrun List<T>)
        public ActionResult ShowNamesOnly()
        {
            var db = new CustomerDataContext();
            var tb = db.Customers;
            List<String> names = (from s in tb
                                  select s.Name).ToList();
            ViewBag.allNames = names;
            return View();
        }
        // immediate query execution ToArray<source> (retrun array i.e [])

        public ActionResult ShowNamesUsingArray()
        {
            var db = new CustomerDataContext();
            var tb = db.Customers;

            String[] names = (from s in tb
                              select s.Name).ToArray();
            ViewBag.allNames = names;
            return View();
        }

        // select name & address
```

```
        public ActionResult nameaddress()
        {
            var db = new CustomerDataContext();

            var tb = db.Customers;

            var q1 = from s in tb
                     select new { s.Name,s.Address };
            ViewBag.nameadd = q1;

            return View();
        }

        // group by method 1

        public ActionResult GroupbyNames()
        {
            var db = new CustomerDataContext();
            var tb = db.Customers;
            var q1 = from s in tb
                     group s by s.Address;

            var res = "";
            foreach (var group in q1)
            {
                res = res + group.Key + ":";
                foreach (var entry in group)
                {
                    res = res + entry.Name + " ";
                }
                res = res + " | ";
            }

            ViewBag.r = res;
            return View();
        }
        //  group by method 2
        public ActionResult GroupbyNamesOnly()
        {
            var db = new CustomerDataContext();
            var tb = db.Customers;
            var q1 = from s in tb
                     group s by s.Address;
            ViewBag.r = q1;
            return View();
        }

        //linq to xml

        public ActionResult XMLData()
        {
            String res = "";
            XDocument xmldoc = XDocument.Load("CustomerDetails.xml");  //using
System.Xml.Linq;
                                                                       // xml file
default path : c:/program files(x86)/iis express
            var q = from c in xmldoc.Descendants("Customer")
                    //select (String)c.Element("CustomerName");
```

```
                    select (String)c.Element("CustomerID") + " - " +
(String)c.Element("CustomerName") + " - " + (String)c.Element("city");

            foreach (String entry in q)
            {
                res = res + entry + " ||";
            }


            ViewBag.ans = res;
            return View();
        }

        public ActionResult Create()
        {
            return View();
        }
        [HttpPost]
        public ActionResult Create(Customer c)
        {
            var db = new CustomerDataContext();
            if (ModelState.IsValid)
            {
                db.Customers.Add(c);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(c);
        }
        public ActionResult Delete(int id)
        {
            var db = new CustomerDataContext();
            Customer c=db.Customers.Find(id);
            if (c == null)
                HttpNotFound();
            return View(c);
        }
        [HttpPost]
        public ActionResult Delete(int id,Customer c)
        {
            var db = new CustomerDataContext();
            Customer fc = db.Customers.Find(id);
            db.Customers.Remove(fc);
            db.SaveChanges();
            if (ModelState.IsValid)
                return RedirectToAction("Index");
            return View(c);
        }
    }
}
```

Model File (Customer.cs)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```
namespace EFwork.Models
{
    public class Customer
    {
        public int ID { get; set; }
        public String Name { get; set; }
        public String Gender { get; set; }
        public String Address { get; set; }
        public String Email { get; set; }
    }
}
```

Model file CutomerDataContext.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.Entity;
namespace EFwork.Models
{
    public class CustomerDataContext:DbContext
    {
        public CustomerDataContext()
            : base("CustomerDBConnection")
        {
        }
        public DbSet<Customer> Customers { get; set; }
    }
}
```

GroupbyNames.cshtml

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>GroupbyNames</title>
</head>
<body>
    <div>
        @ViewBag.r
    </div>
</body>
</html>
```

Groupbynamesonly.cshtml

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>GroupbyNamesOnly</title>
</head>
<body>
    <div>
        @{
            var s = ViewBag.r ;
        }
        <table border="1">
        @foreach (var s1 in s)
        {


            foreach (var a in s1)
            {
                <tr>
                    <td>
                @a.Name
                        </td>
                    </tr>
            }
            <tr><td>. . . . . .</td></tr>

        }
            </table>
    </div>
</body>
</html>
```

NameAddress.cshtml

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>nameaddress</title>
</head>
<body>
    <div>
        @{
            var na = ViewBag.nameadd;
        }
```

```
        @foreach(var s in na)
        {
            @s<br />

        }

    </div>
</body>
</html>
```

Showcount.cshtml

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>ShowCount</title>
</head>
<body>
    <div>
        @{
            var totalRecord = ViewBag.count;
        }
        @totalRecord


        @*  or *@

        Total = @ViewBag.count


    </div>
</body>
</html>
```

showFilterData.cshtml

```
@{
    Layout = null;
}



<!DOCTYPE html>

<html>
```

```
<head>
    <meta name="viewport" content="width=device-width" />
    <title>ShowFilterData</title>
</head>
<body>
    <div>
        @{
            var cus = ViewBag.c;
            }
     @cus<br/>

        @* or *@

        @ViewBag.c
    </div>
</body>
</html>
```

ShownamesOnly.cshtml

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>ShowNamesOnly</title>
</head>
<body>
    <div>
        @{
            var nam = ViewBag.allNames;
        }
        @foreach (var p in nam)
        {
            @p <br />
        }
    </div>
</body>
</html>
```

showNamesusingArray.cshtml

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
```

```
    <meta name="viewport" content="width=device-width" />
    <title>ShowNamesUsingArray</title>
</head>
<body>
    <div>
        @{

            var aNames = ViewBag.allNames;
        }
        @foreach (var s in aNames)
        {
            @s<br />
        }
    </div>
</body>
</html>
```

Xmldata.cshtml

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>XMLData</title>
</head>
<body>
    <div>
        @ViewBag.ans
    </div>
</body>
</html>
```

Web.config file

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
  -->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit
http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework"
type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework,
Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
requirePermission="false" />
  </configSections>
  <connectionStrings>
```

```
    <add name="CustomerDBConnection" connectionString="Data
Source=.\sqlexpress;Integrated Security=SSPI" providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>

……..

……
```

XMLFile(CustomerDetails.xml – file :c:/program files(x86)/iis express/

```
<?xml version="1.0" encoding="utf-8" ?>
- <CustomerDetails>
- <Customer>
  <CustomerID>c001</CustomerID>
  <CustomerName>Gowthaman</CustomerName>
  <city>Karur</city>
    </Customer>
- <Customer>
  <CustomerID>c002</CustomerID>
  <CustomerName>raja</CustomerName>
  <city>chennai</city>
    </Customer>
- <Customer>
  <CustomerID>c003</CustomerID>
  <CustomerName>Raman</CustomerName>
  <city>Namakkal</city>
    </Customer>
- <Customer>
  <CustomerID>c004</CustomerID>
  <CustomerName>Ram</CustomerName>
  <city>Karur</city>
    </Customer>
- <Customer>
  <CustomerID>c005</CustomerID>
  <CustomerName>Sugu</CustomerName>
  <city>velur</city>
    </Customer>
    </CustomerDetails>
```