

# **Scaffolding**

Introduction: so far manually created the controller and views. This requires a lot of time and manual coding. To overcome this problem using Scaffolding.

\*) scaffolding allows you to create auto generated controllers and corresponding views.

\*) scaffolding uses predefined convention for naming controller and views.

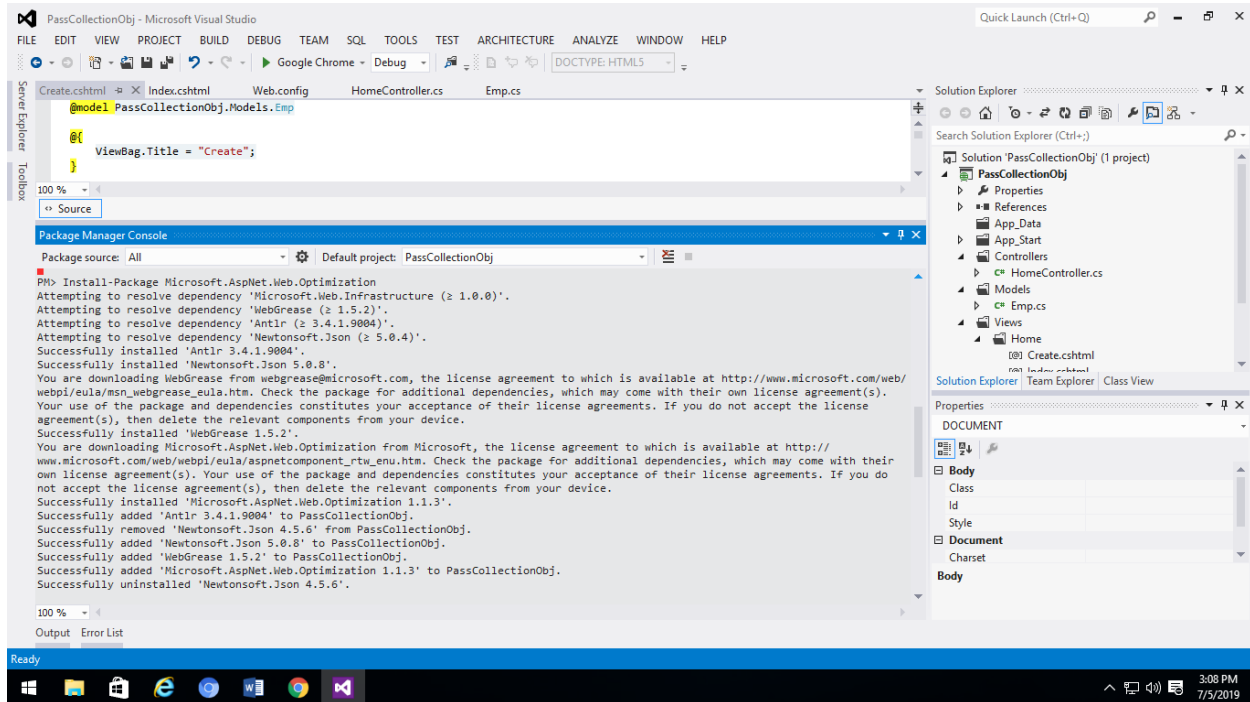
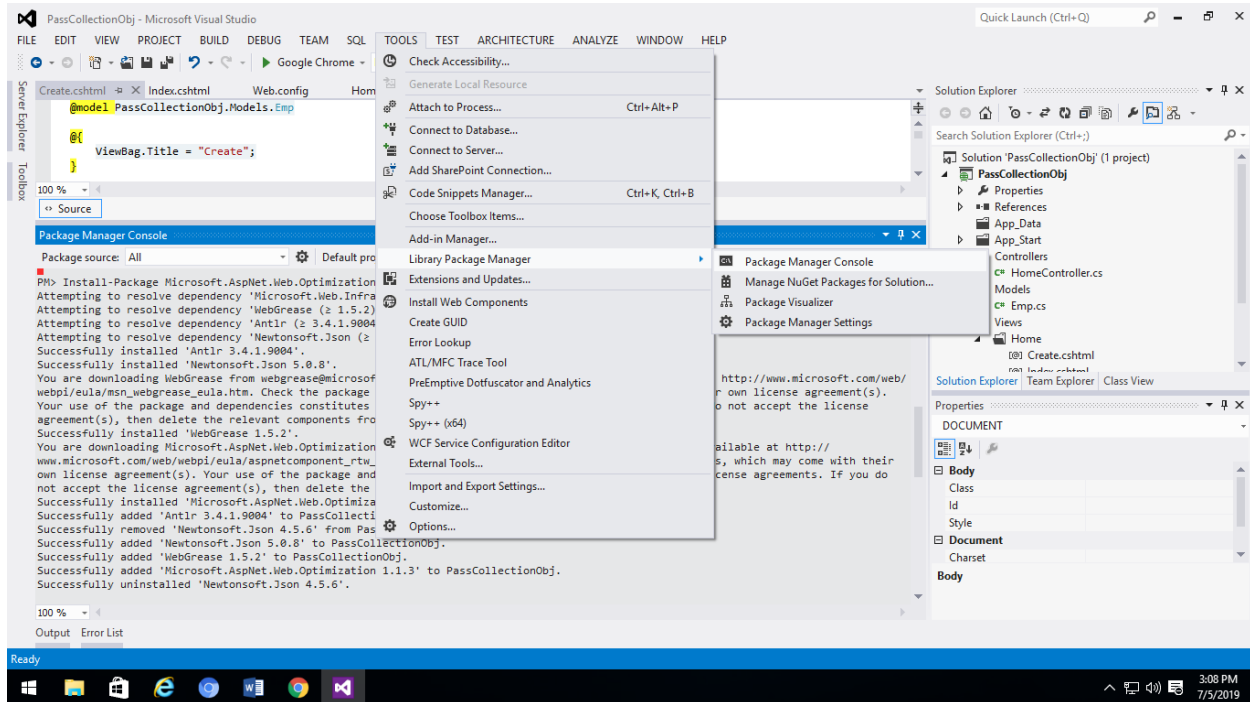
Def.:

Scaffolding is a technique used by many MVC frameworks like ASP.NET MVC, Ruby on Rails, Cake PHP and Node.JS etc., to generate code for basic CRUD (create, read, update, and delete) operations against your database effectively. Further you can edit or customize this auto generated code according to your need.

**Scaffolding provides various templates for creating Controllers and associative views.**

- 1) **Empty MVC controller – derived from Controller class – only one action method index-no code.**
- 2) **MVC controller with empty read/write actions: action method (Index,Details,create,Edit,Delete) – return some code inside them**
- 3) **API Controller with empty read/write action: derived from ApiController**





## Create Scaffold Template.

### Models File(Emp.cs)

```

using System;
using System.Collections.Generic;
  
```

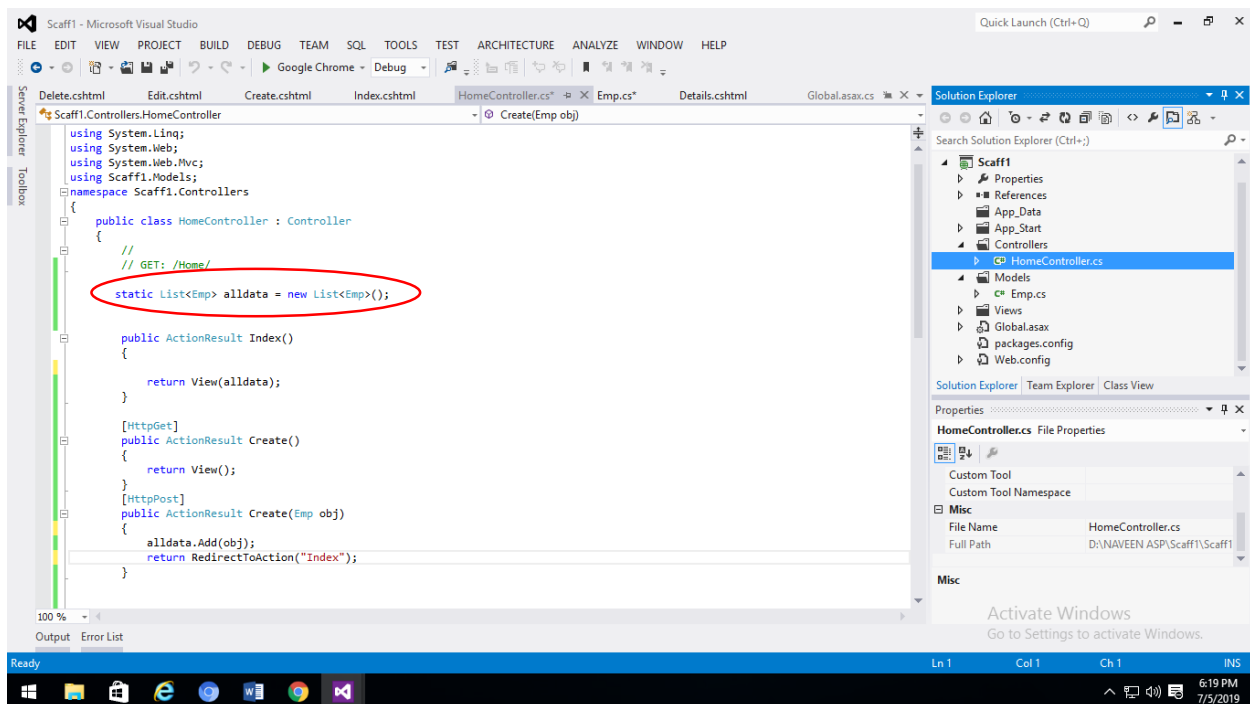
```

using System.Linq;
using System.Web;

namespace Scaff1.Models
{
    public class Emp
    {
        public int eno { get; set; }
        public String ename { get; set; }
        public int esal { get; set; }
    }
}

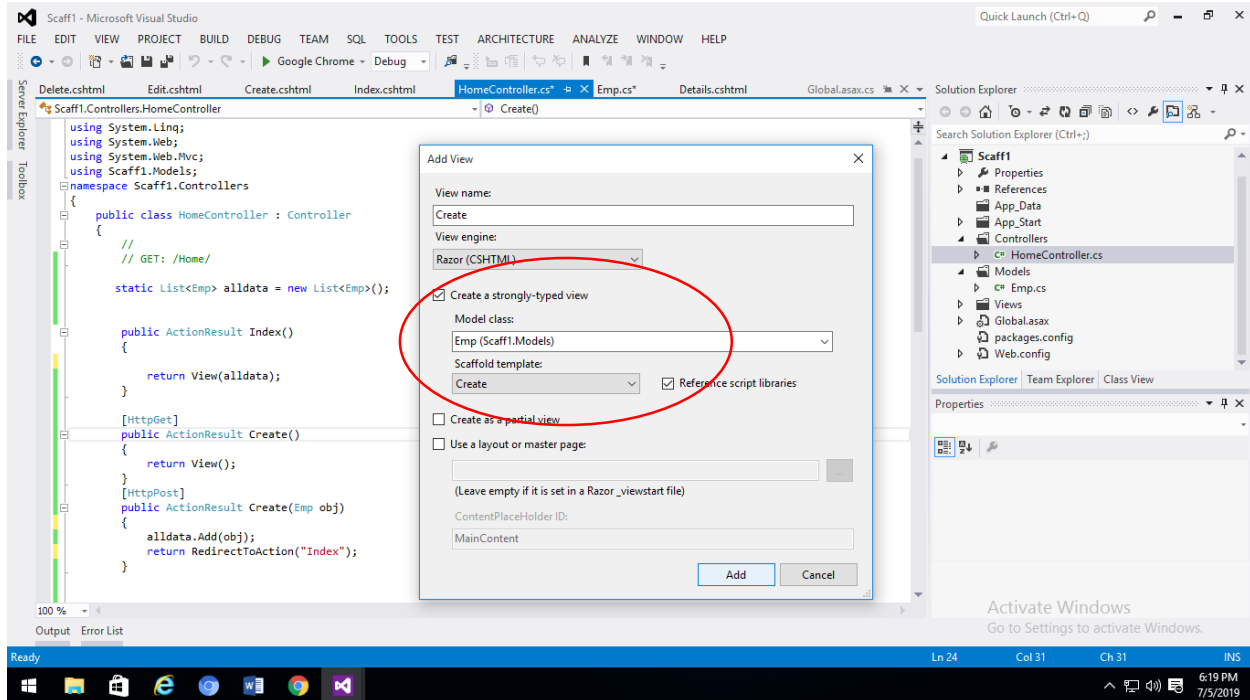
```

Controllers - right click → add controller → scaffolding template → mvc controller with empty read and write action.



1) right click on Index → add view → scaffolding template → List

2) Right click on Create (httpGet) → addview → scaff fold template → create.....add button press.



## View file is automatically Generated(: views ->> home → Create.cshtml)

@model Scaff1.Models.Emp

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Create</title>

</head>

<body>

<script src="~/Scripts/jquery-1.7.1.min.js"></script>

<script src="~/Scripts/jquery.validate.min.js"></script>

<script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

@using (Html.BeginForm()) {

@Html.ValidationSummary(true)

<fieldset>

<legend>Emp</legend>

<div class="editor-label">

@Html.LabelFor(model => model.en)

</div>

<div class="editor-field">

@Html.EditorFor(model => model.en)

@Html.ValidationMessageFor(model => model.en)

</div>

```

        <div class="editor-label">
            @Html.LabelFor(model => model.ename)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.ename)
            @Html.ValidationMessageFor(model => model.ename)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.esal)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.esal)
            @Html.ValidationMessageFor(model => model.esal)
        </div>

        <p>
            <input type="submit" value="Create" />
        </p>
    </fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>
</body>
</html>

```

**Note: when user click submit button(create) → HTTPPost method called at controller file**

**Ex:**

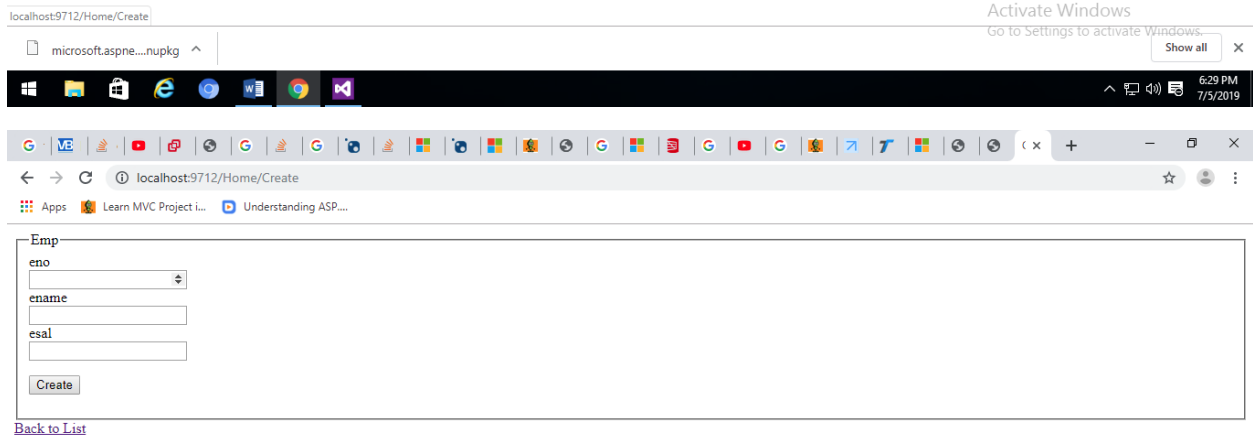
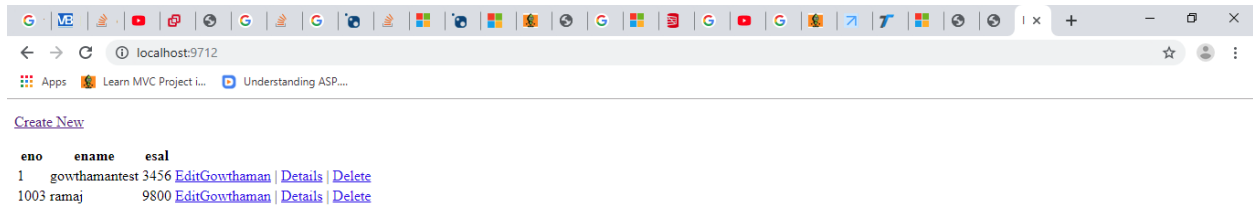
```

[HttpGet] // default no need to type [HttpGet] Annotation
// controller to view
public ActionResult Create()
{
    return View();
}

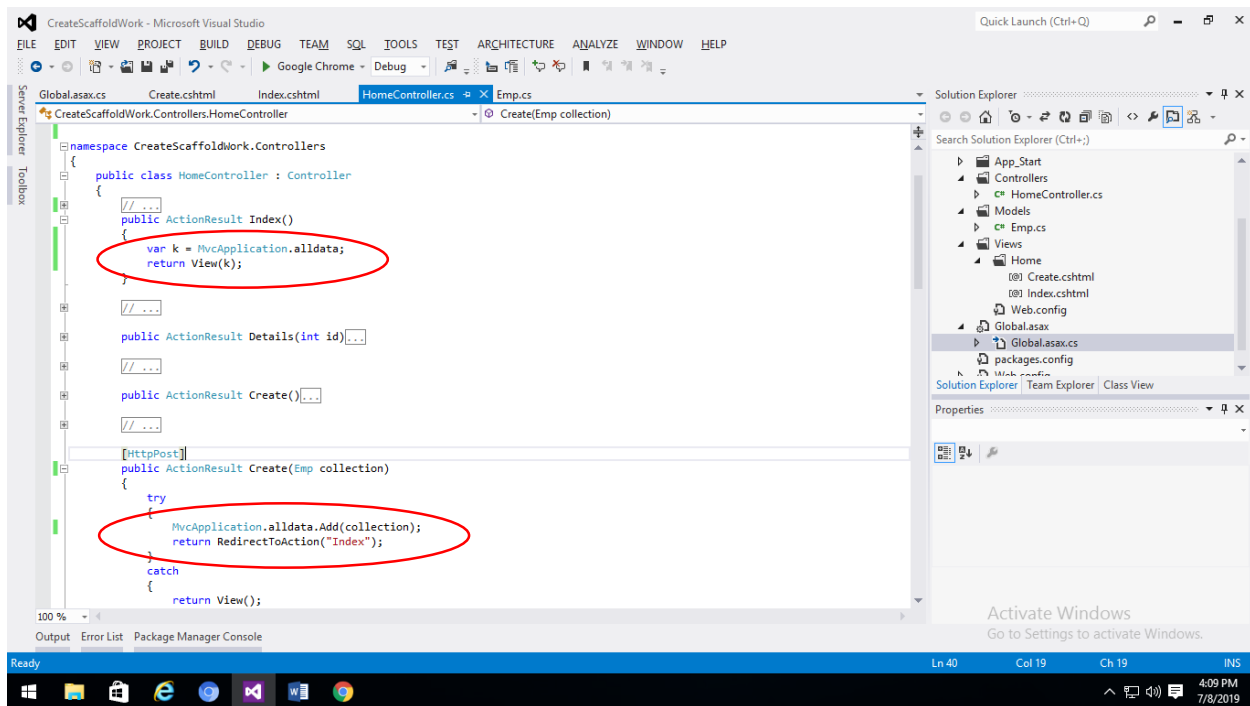
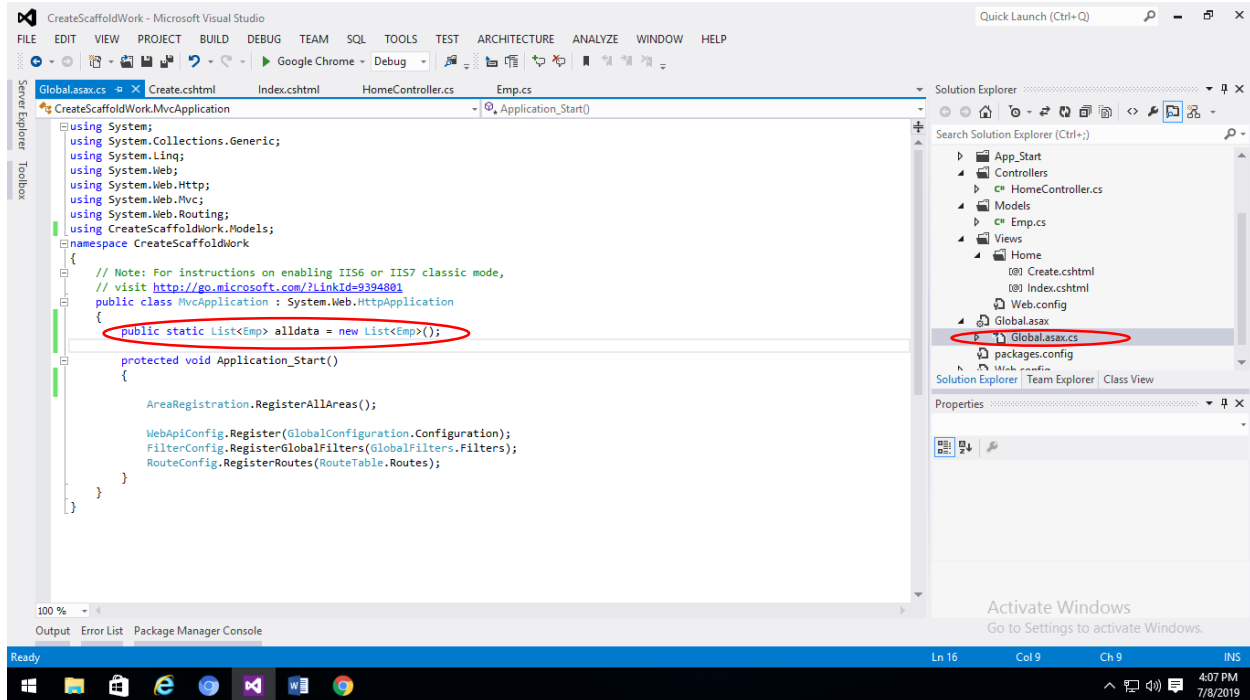
[HttpPost] // view to controller (very important)
public ActionResult Create(Emp obj)
{
    alldata.Add(obj);
    return RedirectToAction("Index");
}

```

**Result:**



Another Method: using Global.asax file

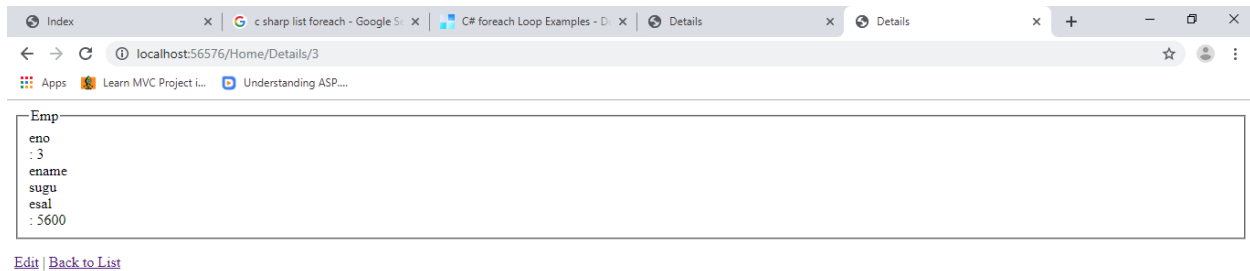




## Details Scaffold work( httpget )

```
public ActionResult Details(int id)
{
    /*Emp e1 = new Emp();
    e1.ename = "gowthaman";
    e1.eno = 199;
    e1.esal = 2000;*/

    var k = MvcApplication.alldata;
    var temp = new Emp();
    foreach (Emp e in k)
    {
        if (e.eno == id)
        {
            temp = e;
            break;
        }
    }
    return View(temp);
}
```

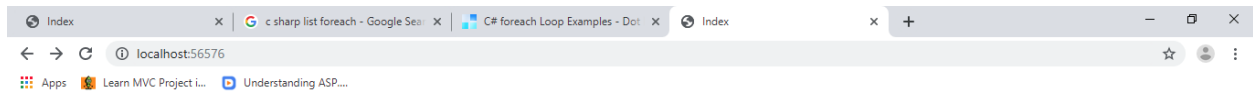


Activate Windows  
Go to Settings to activate Windows.



## Details Scaffold template work( HttpPost )

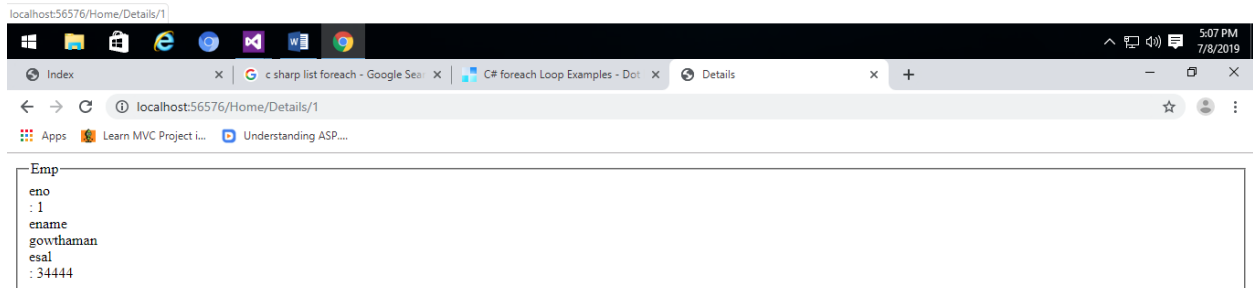
```
[HttpPost]
public ActionResult Details(Emp obj)
{
    return View(obj);
}
```



[Create New](#)

eno	ename	esal	
1	gowthaman	34444	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
2	gowthamantest	234	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

Activate Windows  
Go to Settings to activate Windows.



[Edit](#) | [Back to List](#)

Activate Windows  
Go to Settings to activate Windows.



## Scaffolding Edit:

// GET: /Home/Edit/5

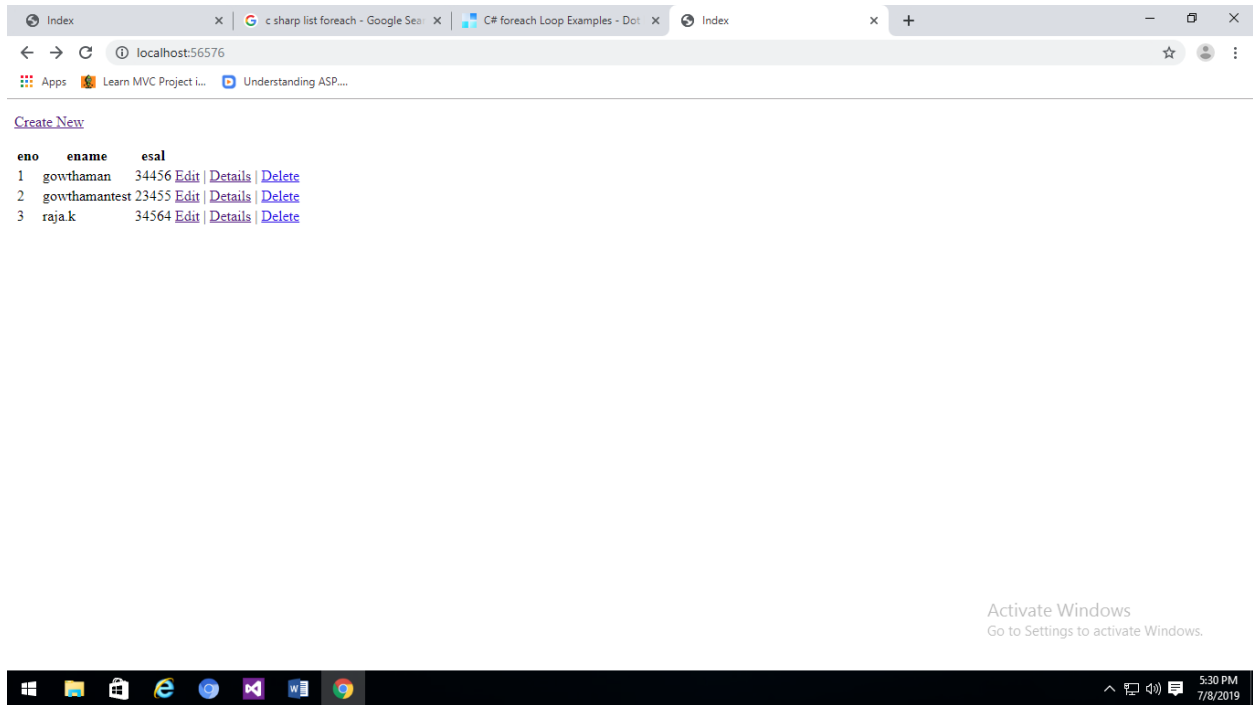
```
public ActionResult Edit(int id)
{
    var k = MvcApplication.alldata;
    var temp = new Emp();
    foreach (Emp e in k)
    {
        if (e.eno == id)
        {
            temp = e;
            break;
        }
    }

    return View(temp);
}

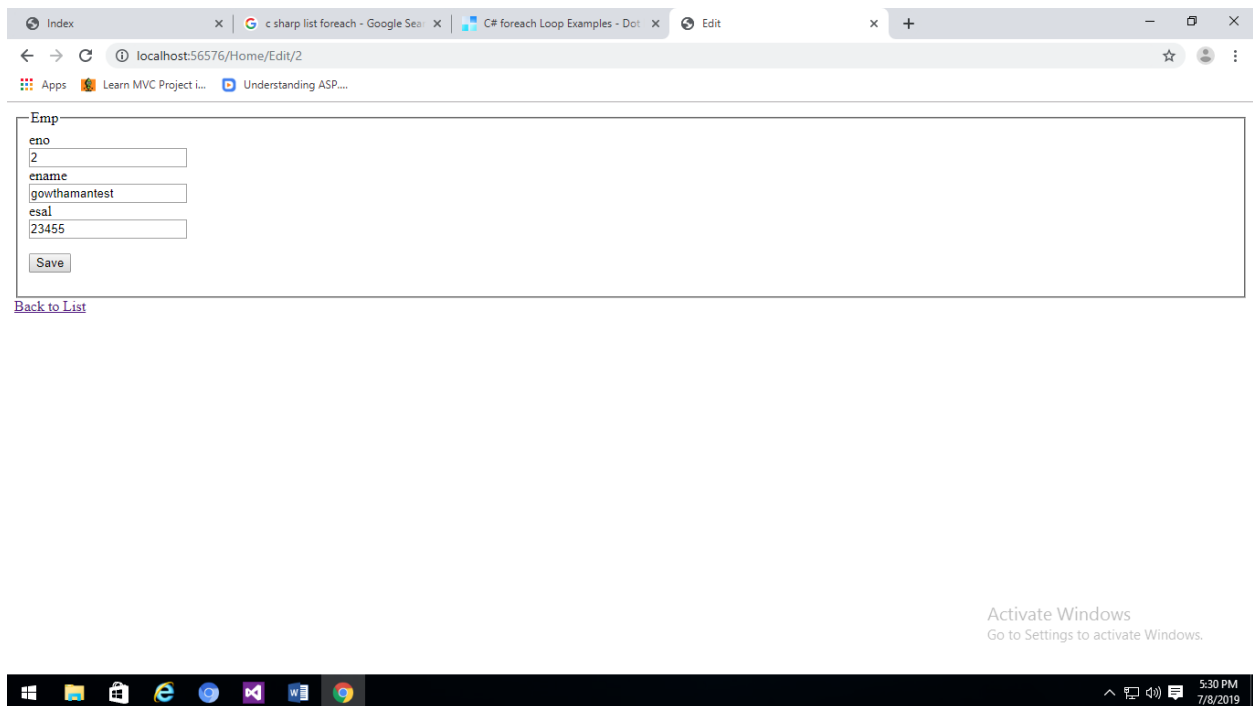
//
// POST: /Home/Edit/5

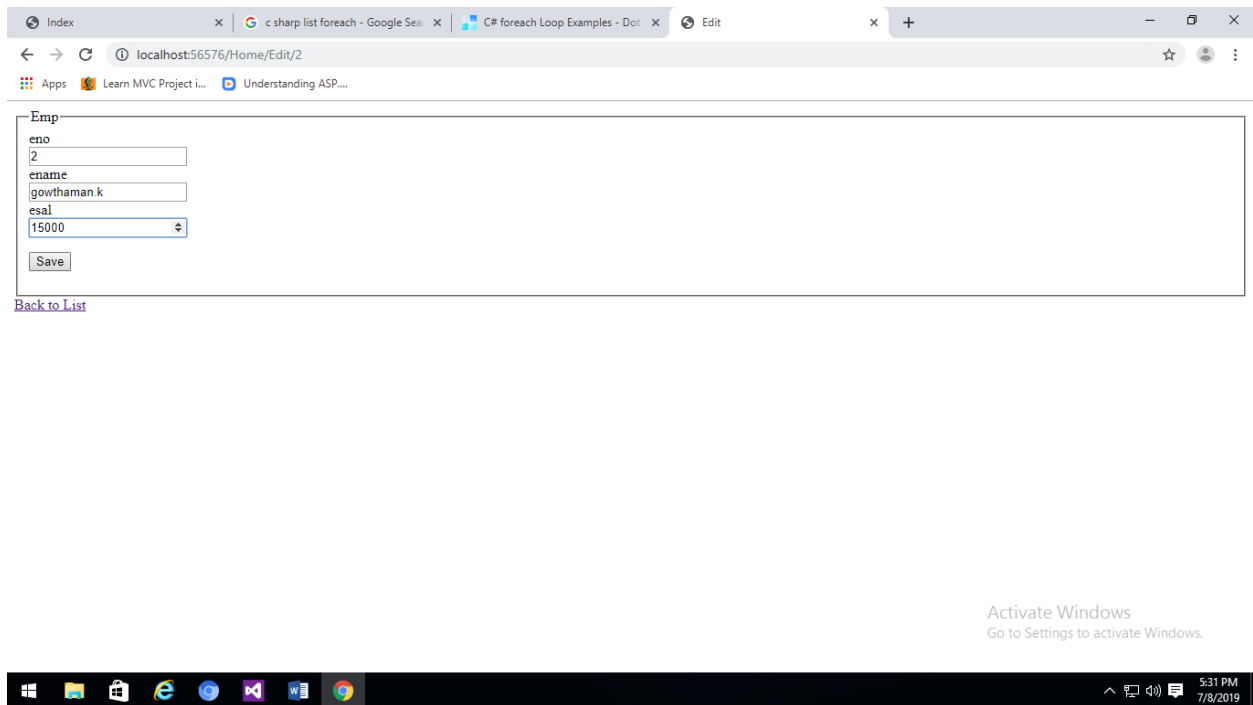
[HttpPost]
public ActionResult Edit(int id, Emp obj)
{
    try
    {
        // TODO: Add update logic here
        var k = MvcApplication.alldata;
        foreach (Emp e in k)
        {
            if (e.eno == id)
            {
                e.ename = obj.ename;
                e.esal = obj.esal;
                break;
            }
        }

        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
```

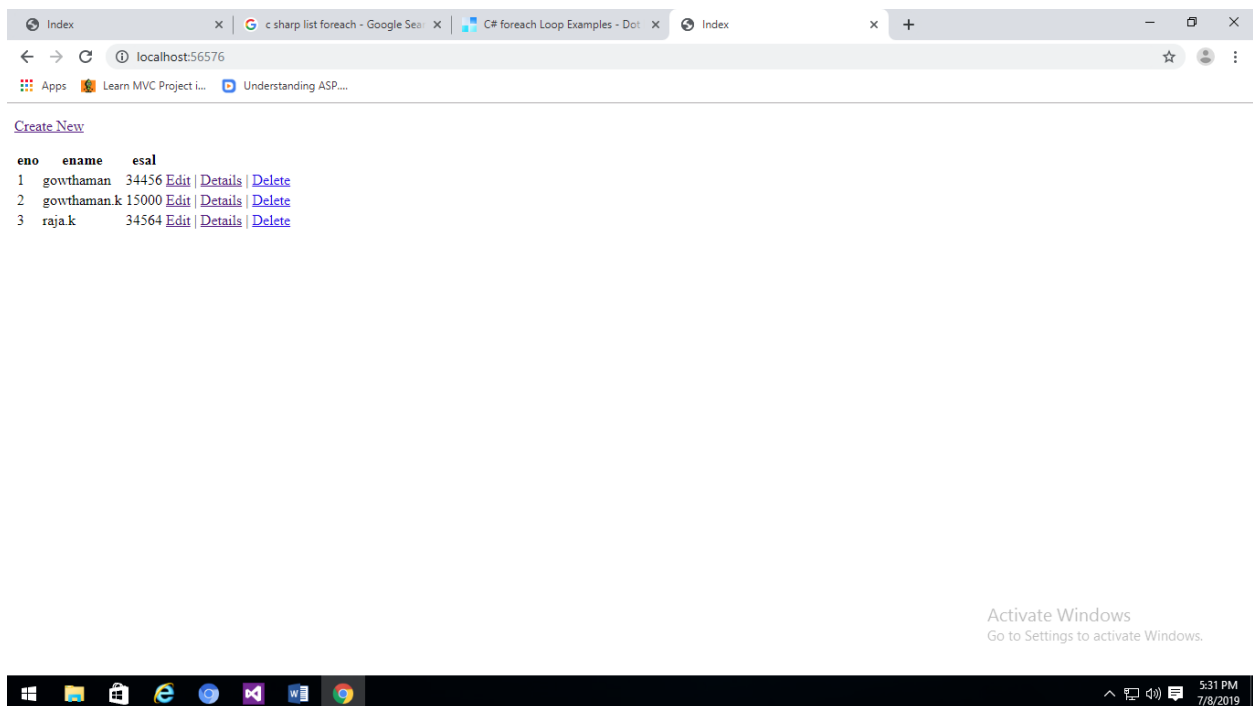


## Press edit link





Finally 2<sup>nd</sup> record updated.



## Scaffolding template Delete Work:

```
// GET: /Home/Delete/5

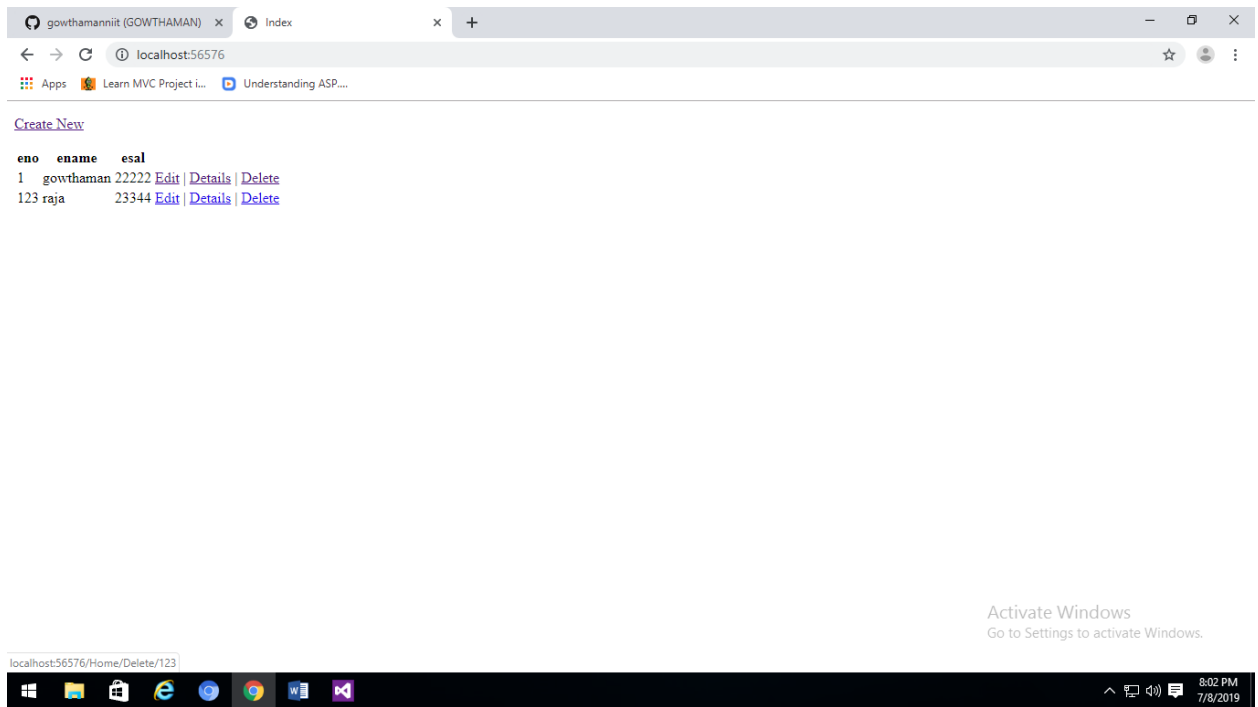
public ActionResult Delete(int id)
{
    var k = MvcApplication.alldata;
    var temp = new Emp();
    foreach (Emp e in k)
    {
        if (e.eno == id)
        {
            temp = e;
            break;
        }
    }
    return View(temp);
}

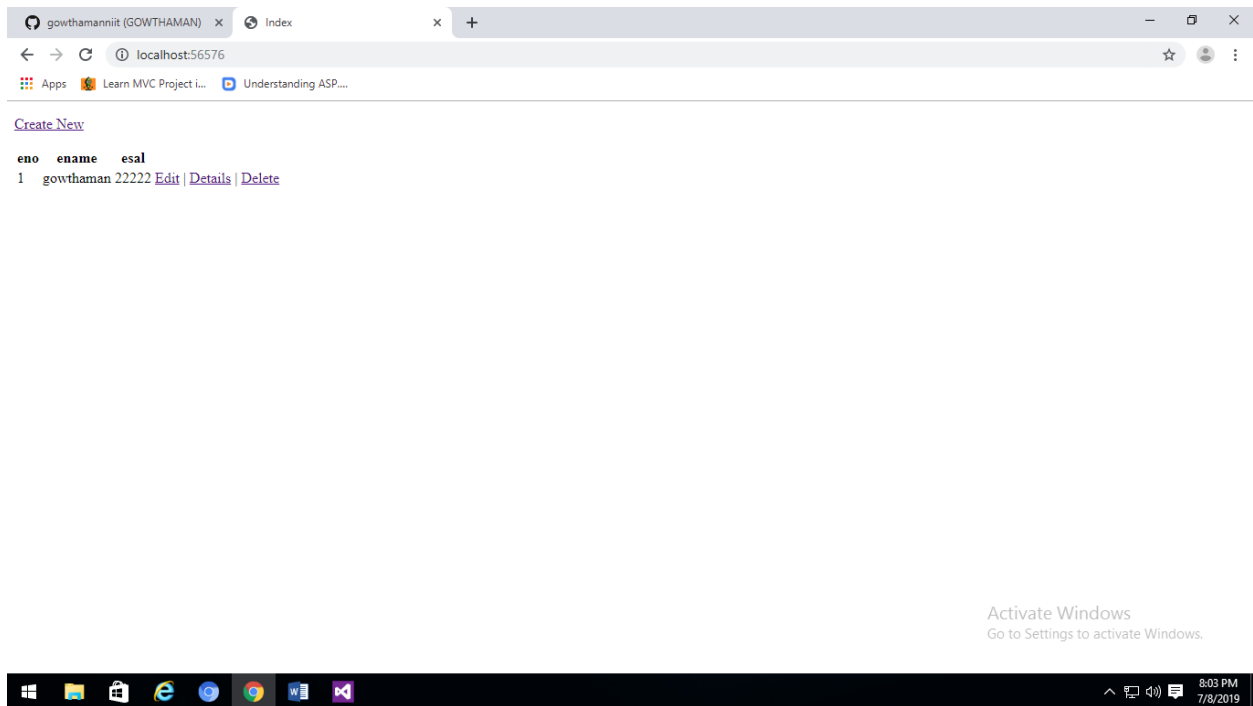
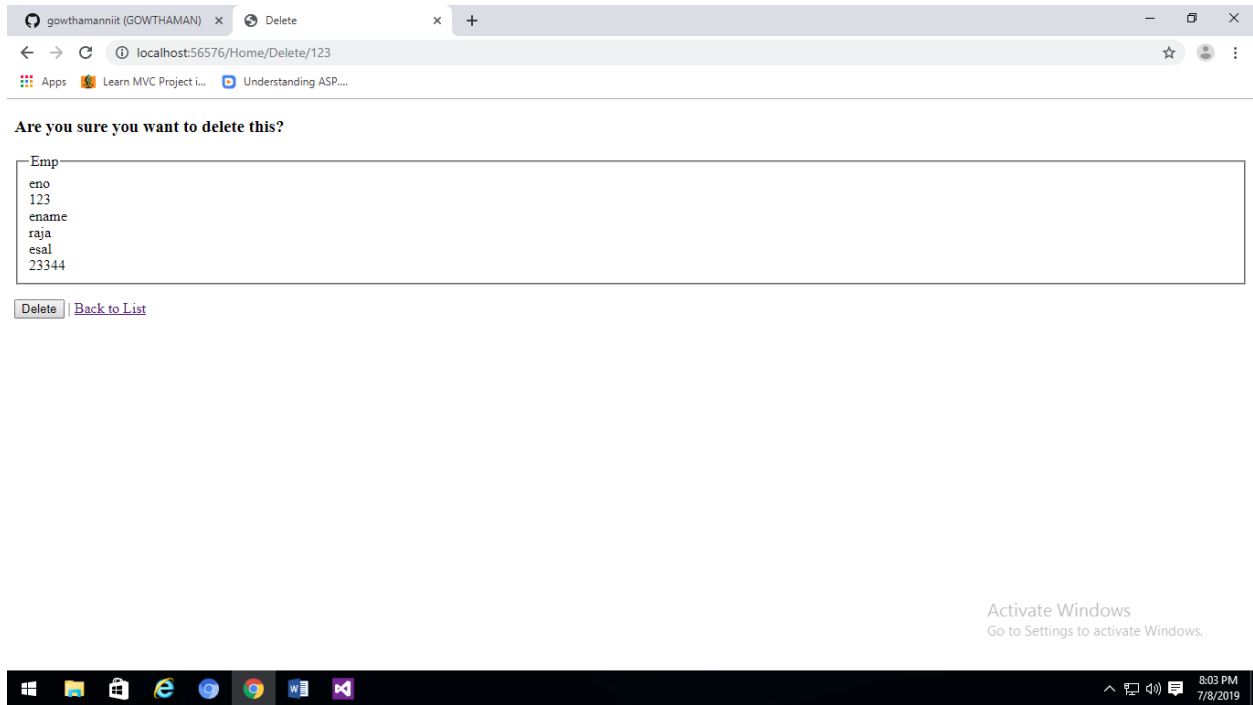
//
// POST: /Home/Delete/5

[HttpPost]
public ActionResult Delete(int id, Emp obj)
{
    try
    {
        // TODO: Add delete logic here
        //MvcApplication.alldata.Clear(); // all data will erase from memory
        // MvcApplication.alldata.Remove(obj); // not work

        var k = MvcApplication.alldata;
        var temp = new Emp();
        foreach (Emp e in k)
        {
            if (e.eno == id)
            {
                k.Remove(e);
                break;
            }
        }

        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
```





## Overall Code:(scaffold all templates)

Home controller:



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using CreateScaffoldWork.Models;

namespace CreateScaffoldWork.Controllers
{
    public class HomeController : Controller
    {
        //
        // GET: /Home/
        // static List<Emp> alldata = new List<Emp>();
        public ActionResult Index()
        {
            var k = MvcApplication.alldata;
            return View(k);
        }

        //
        // GET: /Home/Details/5
        public ActionResult Details(int id)
        {
            /*Emp e1 = new Emp();
            e1.ename = "gowthaman";
            e1.eno = 199;
            e1.esal = 2000;*/

            var k = MvcApplication.alldata;
            var temp = new Emp();
            foreach (Emp e in k)
            {
                if (e.eno == id)
                {
                    temp = e;
                    break;
                }
            }
            return View(temp);
        }

        [HttpPost]
        public ActionResult Details(Emp obj)
        {
            return View(obj);
        }

        //
        // GET: /Home/Create
        public ActionResult Create()
        {
            return View();
        }
    }
}

```

```

//
// POST: /Home/Create

[HttpPost]
public ActionResult Create(Emp collection)
{
    try
    {
        MvcApplication.alldata.Add(collection);
        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}

//
// GET: /Home/Edit/5

public ActionResult Edit(int id)
{
    var k = MvcApplication.alldata;
    var temp = new Emp();
    foreach (Emp e in k)
    {
        if (e.eno == id)
        {
            temp = e;
            break;
        }
    }

    return View(temp);
}

//
// POST: /Home/Edit/5

[HttpPost]
public ActionResult Edit(int id, Emp obj)
{
    try
    {
        // TODO: Add update logic here
        var k = MvcApplication.alldata;
        foreach (Emp e in k)
        {
            if (e.eno == id)
            {
                e.ename = obj.ename;
                e.esal = obj.esal;
                break;
            }
        }
    }
}

```

```

        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}

//
// GET: /Home/Delete/5

public ActionResult Delete(int id)
{
    var k = MvcApplication.alldata;
    var temp = new Emp();
    foreach (Emp e in k)
    {
        if (e.eno == id)
        {
            temp = e;
            break;
        }
    }
    return View(temp);
}

//
// POST: /Home/Delete/5

[HttpPost]
public ActionResult Delete(int id, Emp obj)
{
    try
    {
        // TODO: Add delete logic here
        //MvcApplication.alldata.Clear(); // all data will erase from memory
        // MvcApplication.alldata.Remove(obj); // not work

        var k = MvcApplication.alldata;
        var temp = new Emp();
        foreach (Emp e in k)
        {
            if (e.eno == id)
            {
                k.Remove(e);
                break;
            }
        }

        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
}

```

```
}
```

### Model file

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

namespace CreateScaffoldWork.Models
{
    public class Emp
    {
        public int eno { get; set; }
        public String ename { get; set; }
        public int esal { get; set; }
    }
}
```

### Views (index.cshtml-list scaffold template- view)

```
@model IEnumerable<CreateScaffoldWork.Models.Emp>

@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Index</title>
</head>
<body>
    <p>
        @Html.ActionLink("Create New", "Create")
    </p>
    <table>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.eno)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.ename)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.esal)
            </th>
            <th></th>
        </tr>

        @foreach (var item in Model) {
            <tr>
                <td>
```

```

        @Html.DisplayFor(modelItem => item.eno)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.ename)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.esal)
    </td>
    <td>
        @Html.ActionLink("Edit", "Edit", new { id=item.eno }) |
        @Html.ActionLink("Details", "Details", new { id=item.eno }) |
        @Html.ActionLink("Delete", "Delete", new { id=item.eno })
    </td>
</tr>
}

</table>
</body>
</html>

```

### Edit view (cshtml)

```

@model CreateScaffoldWork.Models.Emp

@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Edit</title>
</head>
<body>
    <script src="~/Scripts/jquery-1.7.1.min.js"></script>
    <script src="~/Scripts/jquery.validate.min.js"></script>
    <script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

    @using (Html.BeginForm()) {
        @Html.ValidationSummary(true)

        <fieldset>
            <legend>Emp</legend>

            <div class="editor-label">
                @Html.LabelFor(model => model.eno)
            </div>
            <div class="editor-field">
                @Html.EditorFor(model => model.eno)
                @Html.ValidationMessageFor(model => model.eno)
            </div>

            <div class="editor-label">
                @Html.LabelFor(model => model.ename)
            </div>

```

```

        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.ename)
            @Html.ValidationMessageFor(model => model.ename)
        </div>

        <div class="editor-label">
            @Html.LabelFor(model => model.esal)
        </div>
        <div class="editor-field">
            @Html.EditorFor(model => model.esal)
            @Html.ValidationMessageFor(model => model.esal)
        </div>

        <p>
            <input type="submit" value="Save" />
        </p>
    </fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>
</body>
</html>

```

### Delete View

```

@model CreateScaffoldWork.Models.Emp

@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Delete</title>
</head>
<body>
    <h3>Are you sure you want to delete this?</h3>
    <fieldset>
        <legend>Emp</legend>

        <div class="display-label">
            @Html.DisplayNameFor(model => model.eno)
        </div>
        <div class="display-field">
            @Html.DisplayFor(model => model.eno)
        </div>

        <div class="display-label">
            @Html.DisplayNameFor(model => model.ename)
        </div>
        <div class="display-field">
            @Html.DisplayFor(model => model.ename)
        </div>
    </fieldset>

```

```

        <div class="display-label">
            @Html.DisplayNameFor(model => model.esal)
        </div>
        <div class="display-field">
            @Html.DisplayFor(model => model.esal)
        </div>
    </fieldset>
    @using (Html.BeginForm()) {
        <p>
            <input type="submit" value="Delete" /> |
            @Html.ActionLink("Back to List", "Index")
        </p>
    }
</body>
</html>

```

### Create view:

```

@model CreateScaffoldWork.Models.Emp

@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Create</title>
</head>
<body>
    <script src="~/Scripts/jquery-1.7.1.min.js"></script>
    <script src="~/Scripts/jquery.validate.min.js"></script>
    <script src="~/Scripts/jquery.validate.unobtrusive.min.js"></script>

    @using (Html.BeginForm()) {
        @Html.ValidationSummary(true)

        <fieldset>
            <legend>Emp</legend>

            <div class="editor-label">
                @Html.LabelFor(model => model.eno)
            </div>
            <div class="editor-field">
                @Html.EditorFor(model => model.eno)
                @Html.ValidationMessageFor(model => model.eno)
            </div>

            <div class="editor-label">
                @Html.LabelFor(model => model.ename)
            </div>
            <div class="editor-field">
                @Html.EditorFor(model => model.ename)
            </div>

```

```

        @Html.ValidationMessageFor(model => model.ename)
    </div>

    <div class="editor-label">
        @Html.LabelFor(model => model.esal)
    </div>
    <div class="editor-field">
        @Html.EditorFor(model => model.esal)
        @Html.ValidationMessageFor(model => model.esal)
    </div>

    <p>
        <input type="submit" value="Create" />
    </p>
</fieldset>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>
</body>
</html>

```

### Details view:

```

@model CreateScaffoldWork.Models.Emp

@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Details</title>
</head>
<body>
    <fieldset>
        <legend>Emp</legend>

        <div class="display-label">
            @Html.DisplayNameFor(model => model.eno)
        </div>
        <div class="display-field">
            : @Html.DisplayFor(model => model.eno)
        </div>

        <div class="display-label">
            @Html.DisplayNameFor(model => model.ename)
        </div>
        <div class="display-field">
            @Html.DisplayFor(model => model.ename)
        </div>

        <div class="display-label">

```

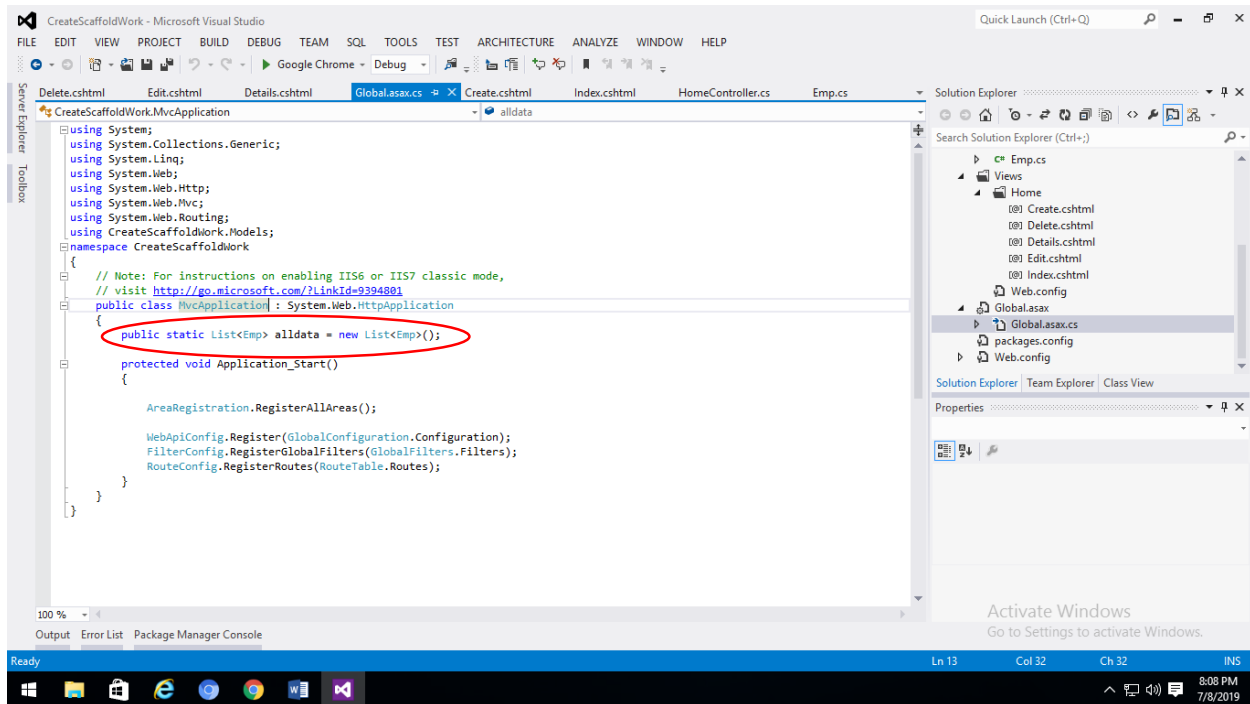


```

        @Html.DisplayNameFor(model => model.esal)
    </div>
    <div class="display-field">
        : @Html.DisplayFor(model => model.esal)
    </div>
</fieldset>
<p>
    @Html.ActionLink("Edit", "Edit", new { /* id=Model.PrimaryKey */ }) |
    @Html.ActionLink("Back to List", "Index")
</p>
</body>
</html>

```

## Global.asax file



<https://www.youtube.com/watch?v=ItSA19x4RU0>

<https://www.youtube.com/watch?v=2WH9FiQLzTY> (create)