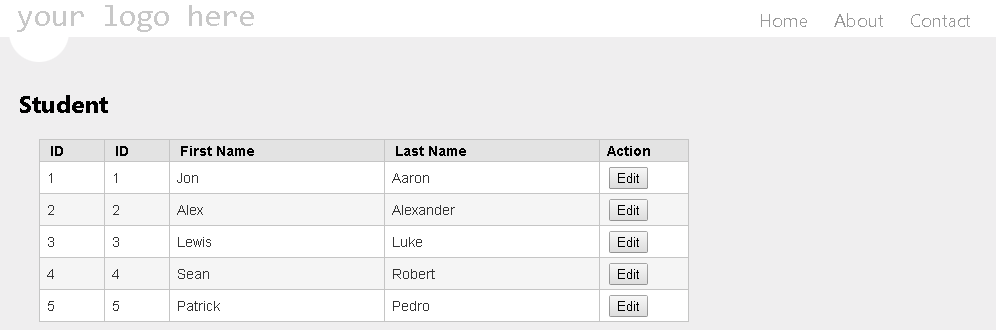
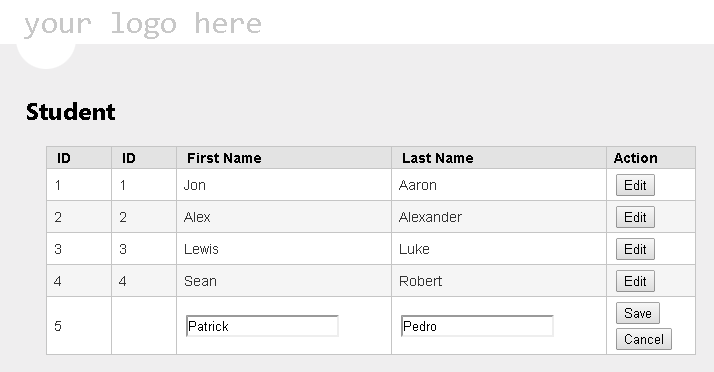
EditableWebgrid

http://www.c-sharpcorner.com/uploadfile/cd7c2e/create-an-editable-webgrid-in-mvc4-to-implement-crud-operati/





@model List<EditableWebgrid.Models.UserModel>

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/\_Layout.cshtml";

}

<style type="text/css">

.edit-mode { }

.edit-user{}

.edit-user display-mode{}

.save-user edit-mode{}

.display-mode{}

.cancel-user{}

.webgrid-table

{

font-family: Arial,Helvetica,sans-serif;

font-size: 14px;

font-weight: normal;

width: 650px;

display: table;

border-collapse: collapse;

border: solid px #C5C5C5;

background-color: white;

}

.webgrid-table td, th

{

border: 1px solid #C5C5C5;

padding: 3px 7px 2px;

}

.webgrid-header, .webgrid-header a

{

background-color: #E3E3E3;

color: black;

text-align: left;

text-decoration:none;

}

.webgrid-footer

{

}

.webgrid-row-style

{

padding: 3px 7px 2px;

}

.webgrid-alternating-row

{

background-color: #F5F5F5;

padding: 3px 7px 2px;

}

.col1Width

{

width: 50px;

}

.col2Width

{

width: 200px;

}

</style>

<script type="text/javascript" >

$(function () {

$('.edit-mode').hide();

$('.edit-user, .cancel-user').on('click', function () {

var tr = $(this).parents('tr:first');

tr.find('.edit-mode, .display-mode').toggle();

});

$('.save-user').on('click', function () {

var tr = $(this).parents('tr:first');

var FirstName = tr.find("#FirstName").val();

var LastName = tr.find("#LastName").val();

var UserID = tr.find("#UserID").html();

tr.find("#lblFirstName").text(FirstName);

tr.find("#lblLastName").text(LastName);

tr.find('.edit-mode, .display-mode').toggle();

var UserModel =

{

"ID": UserID,

"FirstName": FirstName,

"LastName": LastName

};

$.ajax({

url: '/User/UpdateUser/',

data: JSON.stringify(UserModel),

type: 'POST',

contentType: 'application/json; charset=utf-8',

success: function (data) {

alert(data);

}

});

});

})

</script>

<h2>Student</h2>

@{

var grid = new WebGrid(Model); @\*Define your grid\*@

}

<div id="gridContent" style=" padding:20px; " >

@grid.GetHtml(

tableStyle: "webgrid-table",

headerStyle: "webgrid-header",

footerStyle: "webgrid-footer",

alternatingRowStyle: "webgrid-alternating-row",

selectedRowStyle: "webgrid-selected-row",

rowStyle: "webgrid-row-style",

mode: WebGridPagerModes.All,

columns:

grid.Columns(

grid.Column("ID", format: @<text> <span class="display-mode">@item.ID </span> <label id="UserID" class="edit-mode">@item.ID</label> </text>, style: "col1Width" ),

grid.Column("ID", format: @<text> <span class="display-mode">@item.ID </span> </text>, style: "col1Width"),

grid.Column("FirstName", "First Name", format: @<text> <span class="display-mode"> <label id="lblFirstName" >@item.FirstName</label> </span> <input type="text" id="FirstName" value="@item.FirstName" class="edit-mode" /></text>, style: "col2Width"),

grid.Column("LastName", "Last Name", format: @<text> <span class="display-mode"> <label id="lblLastName">@item.LastName</label> </span> <input type="text" id="LastName" value="@item.LastName" class="edit-mode" /> </text>, style: "col2Width"),

grid.Column("Action", format: @<text>

<button class="edit-user display-mode" >Edit</button>

<button class="save-user edit-mode" >Save</button>

<button class="cancel-user edit-mode" >Cancel</button>

</text>, style: "col3Width" , canSort: false)

))

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using EditableWebgrid.Models;

namespace EditableWebgrid.Controllers

{

public class UserController : Controller

{

public ActionResult Index()

{

List<UserModel> users = UserModel.getUsers();

return View(users);

}

public JsonResult UpdateUser(UserModel model)

{

// Update model to your db

string message = "Success";

return Json(message, JsonRequestBehavior.AllowGet);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace EditableWebgrid.Models

{

public class UserModel

{

public int ID { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public static List<UserModel> getUsers()

{

List<UserModel> users = new List<UserModel>()

            {

new UserModel (){ ID=1, FirstName="Jon", LastName="Aaron" },

new UserModel (){ ID=2, FirstName="Alex", LastName="Alexander" },

new UserModel (){ ID=3, FirstName="Lewis", LastName="Luke" },

new UserModel (){ ID=4, FirstName="Sean", LastName="Robert" },

new UserModel (){ ID=5, FirstName="Patrick", LastName="Pedro" },

            };

return users;

}

}

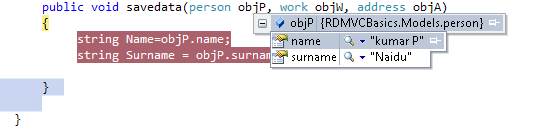
}

**Rendering Layouts Based on a Condition in ASP.Net MVC**

http://www.c-sharpcorner.com/UploadFile/ff2f08/rendering-layouts-base-on-condition-in-Asp-Net-mvc/

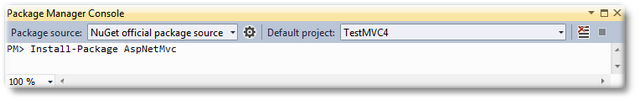
**Bind Data to Multiple Models From View in MVC**

[**http://www.c-sharpcorner.com/UploadFile/dacca2/bind-data-to-multiple-models-from-view-in-mvc/**](http://www.c-sharpcorner.com/UploadFile/dacca2/bind-data-to-multiple-models-from-view-in-mvc/)



http://www.codeproject.com/images/minus.gif Collapse | [Copy Code](http://www.codeproject.com/Articles/468408/ASP-NET-MVC-Setup-Environment)

Install-Package AspNet**Mvc**



Pressing "Enter" key will execute the command,

**Loop through Model items in ASP.NET MVC**

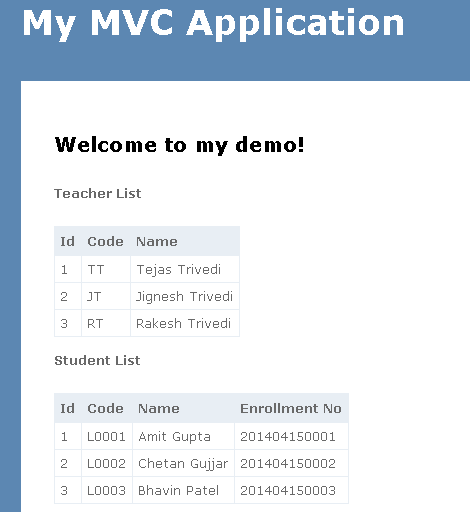
To loop through Model items in ASP.NET MVC view, use foreach loop in the Controller,which returns Collection of items.

Add the given below line to bind your model with .cshtml page.

1. 1: @model IEnumerable<BookStore.Models.Category>

Now write foreach loop at view level page(.cshtml).

1. @**if** (Model != **null**)
2. {
3. **foreach** (var item **in** Model)
4. {
5. <ul id="menu">
7. <li>
8. @Html.DisplayFor(modelItem => item.Address)
9. </li>
10. <li>
11. @Html.DisplayFor(modelItem => item.City)
12. <br />
13. </li>
15. <li>
16. @Html.DisplayFor(modelItem => item.Id)
17. </li>
18. </ul>
19. }
20. }



<http://www.c-sharpcorner.com/UploadFile/ff2f08/multiple-models-in-single-view-in-mvc/>

<http://www.codeproject.com/Articles/687061/Multiple-Models-in-a-View-in-ASP-NET-MVC-MVC>

MultipleModelInOneView

namespace MultipleModelInOneView

{

public class Teacher

{

public int TeacherId { get; set; }

public string Code { get; set; }

public string Name { get; set; }

}

public class Student

{

public int StudentId { get; set; }

public string Code { get; set; }

public string Name { get; set; }

public string EnrollmentNo { get; set; }

}

public class ViewModel

{

public IEnumerable<Teacher> Teachers { get; set; }

public IEnumerable<Student> Students { get; set; }

}

}

IN controller class

public List<Teacher> GetTeachers()

{

List<Teacher> teachers = new List<Teacher>();

teachers.Add(new Teacher { TeacherId = 1, Code = "TT", Name = "Tejas Trivedi" });

teachers.Add(new Teacher { TeacherId = 2, Code = "JT", Name = "Jignesh Trivedi" });

teachers.Add(new Teacher { TeacherId = 3, Code = "RT", Name = "Rakesh Trivedi" });

return teachers;

}

public List<Student> GetStudents()

{

List<Student> students = new List<Student>();

students.Add(new Student { StudentId = 1, Code = "L0001", Name = "Amit Gupta", EnrollmentNo = "201404150001" });

students.Add(new Student { StudentId = 2, Code = "L0002", Name = "Chetan Gujjar", EnrollmentNo = "201404150002" });

students.Add(new Student { StudentId = 3, Code = "L0003", Name = "Bhavin Patel", EnrollmentNo = "201404150003" });

return students;

}

public ActionResult Index()

{

ViewBag.Message = "Welcome to my demo!";

dynamic mymodel = new ExpandoObject();

mymodel.Teachers = GetTeachers();

mymodel.Students = GetStudents();

return View(mymodel);

}

## Creating PDF with ASP.Net MVC and RazorPDF

http://www.dotnetjalps.com/2013/06/Creating-PDF-with-ASP-Net-MVC-and-RazorPDF.html