

Bài 7: *Hiển thị danh sách dữ liệu bằng Razor trong ASP.NET MVC*

Trong bài này mình sẽ hướng dẫn cho các bạn cách để liệt kê danh sách dữ liệu từ 1 bảng ra giao diện người dùng. Đồng thời tùy biến cách hiển thị. Bài này chúng ta cần tóm tắt các bước sau:

- Generate lại tầng Entity Framework Code First theo các bảng mới.
- Tạo controller và model để đọc dữ liệu từ thủ tục trả về.
- Tạo View để hiển thị dữ liệu

Thực hiện:

Chuẩn bị dữ liệu

Tạo bảng Product:

```
USE [WebShopHocTap]
GO
```

```
/****** Object:  Table [dbo].[Product]          Script Date: 03/14/2018
22:01:04 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Product] (
    [ID] [int] NOT NULL,
    [Name] [nvarchar] (50) NULL,
    [Alias] [varchar] (50) NULL,
    [CategoryID] [int] NULL,
    [Images] [nvarchar] (50) NULL,
    [CreateDate] [datetime] NULL,
    [Price] [decimal] (18, 0) NULL,
    [Detail] [ntext] NULL,
    [Status] [bit] NULL,
    CONSTRAINT [PK_Product] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
Tạo Bảng Category
USE [WebShopHocTap]
GO

/***** Object:  Table [dbo].[Category]      Script Date: 03/14/2018
22:02:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Category] (
    [ID] [INT] NOT NULL,
    [Name] [NVARCHAR] (50) NULL,
    [Alias] [VARCHAR] (50) NULL,
    [ParentID] [INT] NULL,
    [CreateDate] [DATETIME] NULL,
    [Order] [INT] NULL,
    [Status] [BIT] NULL,
    CONSTRAINT [PK_Category] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
```

Chuẩn bị Entity Framework

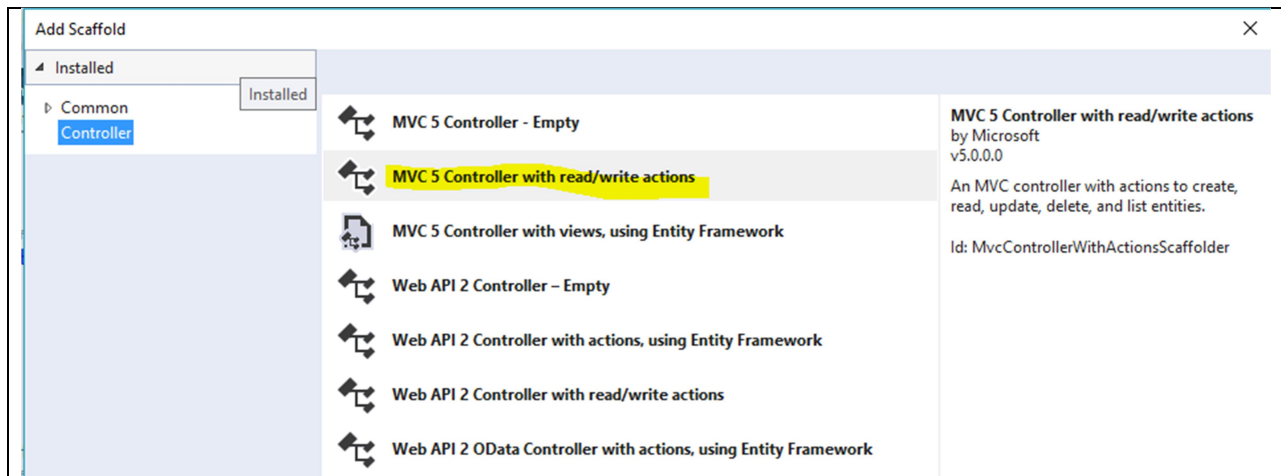
Khi có sự thay đổi về cấu trúc của database (thay đổi table, view) thì cần phải thực hiện tạo lại Entity Framework mới cho project.

(các bước làm tương tự như trong bài thực hành 6)

Tạo controller và model để đọc dữ liệu từ thủ tục trả về

Task 1: hiển thị danh sách danh mục sản phẩm

Bước 1: Tạo 1 controller có tên là CategoryController



Sau khi tạo ta có một controller mới với các phương thức read write trống.

Bước 2: trong Project Models, tạo Class CategoryModel

```
using Models.Framework;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Models
{
    public class CategoryModel : IDisposable
    {
        private OnlineShopDbContext context = null;
        public void Dispose()
        {
            if (context != null)
            {
                context.Dispose();
                context = null;
            }
        }
        public CategoryModel()
        {
            context = new OnlineShopDbContext();
        }
        public List<Category> ListAll()
        {
            var list =
context.Database.SqlQuery<Category>("SP_Category_ListAll").ToList();
            return list;
        }
    }
}
```

Thủ tục lấy dữ liệu category trong sql

```
CREATE PROC SP_Category_ListAll
AS
    SELECT ID ,
           Name ,
           Alias ,
           ParentID ,
           CreateDate ,
           Orders ,
           Status
    FROM Category
    WHERE Status = 1
    ORDER BY Orders ASC;
```

Trong CategoryController thực hiện gọi view

```
public class CategoryController : Controller
{
    //
    // GET: /Admin/Category/
    public ActionResult Index()
    {
        var ipcate = new CategoryModel();
        var model = ipcate.ListAll();
        return View(model);
    }
}
```

Tạo view cho Action này.

Add View

View name:

Index

Template:

List

Model class:

Category (Models.Framework)

Data context class:

OnlineShopDbContext (Models.Framework)

View options:

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page:

~/Areas/Admin/Views/Shared/_Layout.cshtml

...

(Leave empty if it is set in a Razor _viewstart file)

Add

Cancel

```
@model IEnumerable<Models.Framework.Category>
```

```
@{  
    ViewBag.Title = "Index";  
    Layout = "~/Areas/Admin/Views/Shared/_Layout.cshtml";  
}
```

```
<h2>Danh sách Category</h2>
```

```
<p>  
    @Html.ActionLink("Create New", "Create")  
</p>  
<table class="table">  
    <tr>  
        <th>  
            @Html.DisplayNameFor(model => model.Name)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.Alias)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.ParentID)  
        </th>  
        <th>
```

```

        @Html.DisplayNameFor(model => model.CreateDate)
    </th>
    <th>
        @Html.DisplayNameFor(model => model.Order)
    </th>
    <th>
        @Html.DisplayNameFor(model => model.Status)
    </th>
    <th></th>
</tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @item.Name
            @* @Html.DisplayFor(modelItem => item.Name)*@
        </td>
        <td>
            @item.Alias
            @* @Html.DisplayFor(modelItem => item.Alias)*@
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.ParentID)
        </td>
        <td>
            @(item.CreateDate.HasValue?item.CreateDate.Value.ToShortDateString():"")
            @*@Html.DisplayFor(modelItem =>
item.CreateDate.Value.ToShortDateString())*@
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Order)
        </td>
        <td>
            @(item.Status.Value?"Còn sử dụng":"Đã xóa")
            @*@Html.DisplayFor(modelItem => item.Status)*@
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id=item.ID }) |
            @Html.ActionLink("Details", "Details", new { id=item.ID }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.ID })
        </td>
    </tr>
}
</table>

```

Có thể sử dụng 1 template hiển thị trong view này để hiển thị dữ liệu