

Bài 11: Update, Delete dữ liệu với ASP.NET MVC và Entity Framework

Mục tiêu:

- Tạo view sửa;
- Viết model và thủ tục truy xuất cơ sở dữ liệu
- Validate form
- Cấu tạo thành phần của Views

Thủ tục để update dữ liệu

```
USE [WebShopHocTap]
GO
/***** Object:  StoredProcedure [dbo].[SP_Category_Insert]      Script Date:
03/22/2018 00:33:03 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROC [dbo].[SP_Category_Update]
@ID INT,
@Name NVARCHAR(50),
@Alias VARCHAR(50),
@ParentID INT,
@Order INT,
@STATUS bit
AS
BEGIN
    UPDATE dbo.Category
    set
        Name =@Name,
        Alias =@Alias,
        ParentID=@ParentID ,
        CreateDate=GETDATE() ,
        [Order]=@Order ,
        [Status]=@STATUS

    WHERE ID=@ID
END
```

Trong lớp CategoryModel của project Model thêm các phương thức

```
//Lấy một danh mục Category theo ID
public Category GetByID(int id)
{
    object[] parameter={
        new SqlParameter("@ID",id)
    };
}
```

```

        var list =
context.Database.SqlQuery<Category>("SP_Category_GetByID @ID",
parameter).SingleOrDefault();
        return list;
    }
    //Phương thức update Category
    public int UpdateCategory(int id, string name, string alias, int?
parentID, int? order, bool? status)
    {
        object[] parameter =
        {
            new SqlParameter("@ID", id),
            new SqlParameter("@Name", name),
            new SqlParameter("@Alias", alias),
            new SqlParameter("@ParentID", parentID),
            new SqlParameter("@Order", order),
            new SqlParameter("@Status", status)
        };
        int res =
context.Database.ExecuteSqlCommand("SP_Category_Update
@ID,@Name,@Alias,@ParentID,@Order,@Status", parameter);
        return res;
    }

```

Trong lớp CategoryController thêm 2 Action

```

//Action để hiển thị dữ liệu khi click vào edit
// GET: /Admin/Category/Edit/5

    public ActionResult Edit(int id)
    {
        var result = new CategoryModel();
        var model = result.GetByID(id);
        return View(model);
    }

    //
    //Action sử dụng để sửa thông tin
    // POST: /Admin/Category/Edit/5
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit(int id, Category collection)
    {
        try
        {
            if (ModelState.IsValid)

```

```

        {
            // TODO: Add insert logic here
            var result = new CategoryModel();
            int res = result.UpdateCategory(collection.ID,
collection.Name, collection.Alias, collection.ParentID, collection.Order,
collection.Status);
            if (res > 0)
            {
                return RedirectToAction("Index");
            }
            else
            {
                ModelState.AddModelError("", "Không thêm được sản phẩm");
            }
        }
        return View(collection);
    }
    catch
    {
        return View();
    }
}

```

Tạo view Edit giống như view Create

Có thể copy view Create rồi paste vào view Edit

@model Models.Framework.Category

@{

 ViewBag.Title = "Update";
 Layout = "~/Areas/Admin/Views/Shared/_Layout.cshtml";

}

<h2>Update</h2>

@using (Html.BeginForm())

{

 @Html.AntiForgeryToken()

 <div class="form-horizontal">

 <h4>Category</h4>

 <hr />

 @Html.ValidationSummary(true)

 <div class="form-group">

 @Html.LabelFor(model => model.ID, new { @class = "control-label col-md-2" })

 <div class="col-md-10">

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

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

 </div>

 </div>

```
<div class="form-group">
    @Html.LabelFor(model => model.Name, new { @class = "control-label
col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Name)
        @Html.ValidationMessageFor(model => model.Name)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Alias, new { @class = "control-
label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Alias)
        @Html.ValidationMessageFor(model => model.Alias)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.ParentID, new { @class = "control-
label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.ParentID)
        @Html.ValidationMessageFor(model => model.ParentID)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.CreateDate, new { @class =
"control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.CreateDate)
        @Html.ValidationMessageFor(model => model.CreateDate)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Order, new { @class = "control-
label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Order)
        @Html.ValidationMessageFor(model => model.Order)
    </div>
</div>

<div class="form-group">
    @Html.LabelFor(model => model.Status, new { @class = "control-
label col-md-2" })
    <div class="col-md-10">
        @Html.DropDownListFor(model => model.Status, new
```

```
List<SelectListItem>()
{
    new SelectListItem(){Selected=true,Text="kích hoạt",Value="True"},
    new SelectListItem(){Selected=false,Text="đóng",Value="false"}
}, new { @class = "form-control" })

    </div>
</div>

<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Update" class="btn btn-default"
/>
    </div>
</div>
</div>
}

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

Thực hiện xóa dữ liệu bằng microsoft jquery unobtrusive ajax

Thực hiện thủ tục xóa

```
CREATE PROC SP_Category_Delete
@ID int
AS
DELETE dbo.Category
WHERE ID=3
```

Viết hàm xóa trong CategoryModel

```
public int DeleteCategoryByID(int id)
{
    object[] parameter ={
        new SqlParameter("@ID",id)
    };
    var res = context.Database.ExecuteSqlCommand("SP_Category_Delete
@ID", parameter);
    return res;
}
```

Viết action Delete trong CategoryController

```
[HttpDelete]
public ActionResult Delete(int id)
{
    try
    {
        // TODO: Add insert logic here
        var result = new CategoryModel();
        int res = result.DeleteCategoryByID(id);
    }
}
```

```
        if (res > 0)
        {
            return RedirectToAction("Index");
        }
        else
        {
            ModelState.AddModelError("", "Không Xóa được sản phẩm");
        }
        return View("Index");
    }
    catch (Exception ex)
    {
        return View("Index");
    }
}
```

Thực hiện sử dụng Ajax trong view Index.

```
@Ajax.ActionLink("Xóa", "Delete", new { @id=item.ID }
, new AjaxOptions
{
    Confirm="Bạn có muốn xóa bản ghi này?",
    OnComplete = string.Format("${'#row_{0}'} .remove()",@item.ID),
    HttpMethod = "Delete"
})
```

Muốn thực hiện được phải cài microsoft jquery unobtrusive ajax sau đó kéo vào trong _Layout.cshtml sau:

```
<script src="/Assets/Admin/vendor/jquery/jquery.min.js"></script>
<script src="/Scripts/jquery.unobtrusive-ajax.min.js"></script>
```