# Bài thực hành 8: xây dựng trang quản trị cho TOPICS

# Menu trash

### Thiết kế chức năng:



Khi nhấn vào biểu tượng thùng rác sẽ hiển thị các mẩu tin đã xóa (Status = 0).

Và có 2 chức năng: Phục hồi (chuyển trạng thái về trạng thái =2), xóa hẳn (xóa ra khỏi cơ sở dữ liệu)

## Tạo controller cho trash

Để hiển thị thùng rác nghĩa là xem xét các mẩu tin có Status = 0. Hàm Getlist trong TopicDAO

public List<Topics> getList(string status = "All")

{

List<Topics> list = null;

switch (status)

{

case "Index":

{

list = db.Topics

.Where(m => m.Status != 0)

.ToList();

break;

}

case "Trash":

{

list = db.Topics

.Where(m => m.Status == 0)

.ToList();

break;

}

default:

{

list = db.Topics.ToList();

break;

}

}

return list;

}

#### Để ý thầy răng Status mặc định là All, ta sẽ truyền status = “Trash”

Sao chép nội dung của Index

/////////////////////////////////////////////////////////////////////////////////////

// GET: Admin/Topic/Trash/5:Hien thi cac mau tin có gia tri la 0

public ActionResult Trash(int? id)

{

return View(topicDAO.getList("Trash"));

}

## Tạo giao diện trang thùng rác

R\_Click Trash - Add View

Empty (Empty Model)

Name = Trash

### Chỉnh sửa tiêu đề

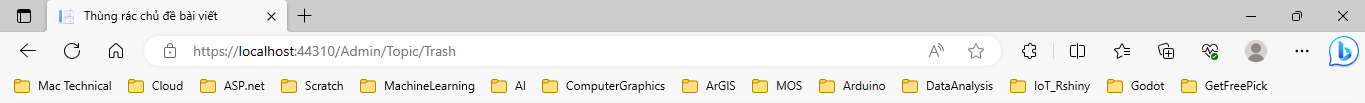
@model IEnumerable<MyClass.Model.Topics>

@{

ViewBag.Title = "Thùng rác chủ đề bài viết";

Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}



### Chỉnh sửa giao diện

Danh sách hiển thị giống với trang Index (Copy tòa bộ trang Index bỏ vô)

Lưu ý: Khai báo sử dụng Model Topics

@model IEnumerable<MyClass.Model.Topics>

Và hiển thị dưới dạng bảng biểu

@section header {

<link href="~/Public/css/cdn.datatables.net\_1.13.6\_css\_jquery.dataTables.min.css" rel="stylesheet" />

}

@section footer {

<script src="~/Public/js/cdn.datatables.net\_1.13.6\_js\_jquery.dataTables.min.js"></script>

<script>let table = new DataTable('#myTable');</script>

}

### Menu tiêu đề

Chỉ có nút quay trở về trang Index (tham khảo Details)

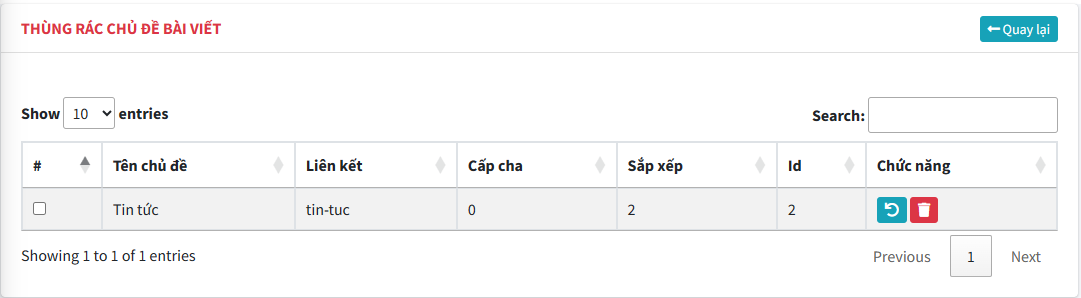
<a href="@Url.Action("Index","Topic")" class="btn-sm btn-info"> <i class="fa fa-long-arrow-alt-left"></i> Quay lại</a>

### Menu chức năng

Chỉ có nút Recover và Delete (Xóa khỏi cơ sở dữ liệu)

<a href="@Url.Action("Recover","Supplier", new { id = item.Id })" class="btn-sm btn-info"> <i class="fa fa-undo"></i></a>

<a href="@Url.Action("Delete","Supplier", new { id = item.Id })" class="btn-sm btn-danger"> <i class="fa fa-trash"></i></a>



# phục hồi dữ liệu

## Thiết lập hàm recover

Hàm Recover chính là việc chuyển trạng thái Status = 0 sang = 2 (Tham khảo chức năng DelTrash).

Lưu ý: sau khi phục hồi tập tin thì vẫn ở trong thùng rác.

#### (yêu cầu hơn: Chuyển về trạng thái trước khi xóa?)

/////////////////////////////////////////////////////////////////////////////////////

// GET: Admin/Topic/Recover/5:Thay doi trang thai cua mau tin

public ActionResult Recover(int? id)

{

if (id == null)

{

//Thong bao that bai

TempData["message"] = new XMessage("danger", "Phục hồi chủ đề thất bại");

//chuyen huong trang

return RedirectToAction("Index", "Topic");

}

//khi nhap nut thay doi Status cho mot mau tin

Topics topics = topicsDAO.getRow(id);

//kiem tra id cua topics co ton tai?

if (topics == null)

{

//Thong bao that bai

TempData["message"] = new XMessage("danger", "Phục hồi chủ đề thất bại");

//chuyen huong trang

return RedirectToAction("Index", "Topic");

}

//thay doi trang thai Status = 2

topics.Status = 2;

//cap nhat gia tri cho UpdateAt/By

topics.UpdateBy = Convert.ToInt32(Session["UserId"].ToString());

topics.UpdateAt = DateTime.Now;

//Goi ham Update trong ProductDAO

topicsDAO.Update(topics);

//Thong bao thanh cong

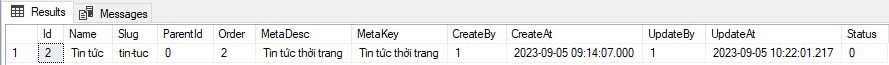
TempData["message"] = new XMessage("success", "Phục hồi sản phẩm thành công");

//khi cap nhat xong thi chuyen ve Trash de phuc hoi tiep

return RedirectToAction("Trash", "Topic");

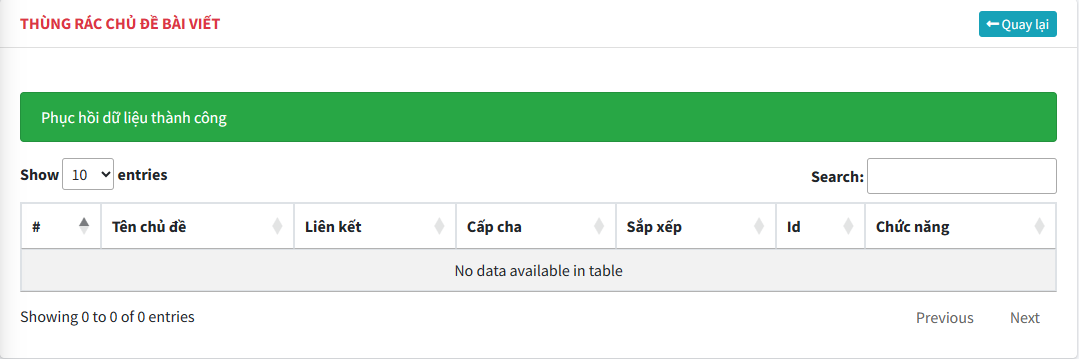
}

### Trước khi phục hồi

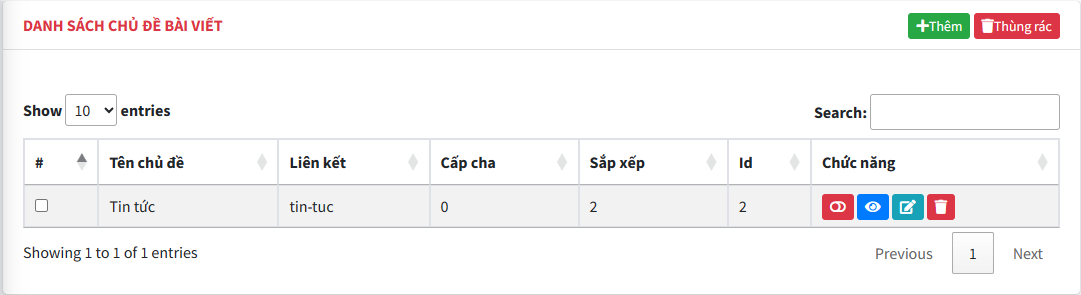


### 

### Sau khi phục hồi



#### Status = 2



# xóa ra khỏi cơ sở dữ liệu

Hàm Delete được xây dựng ngay từ đầu bởi EntityFrameWork

/////////////////////////////////////////////////////////////////////////////////////

// GET: Admin/Topic/Delete/5

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Topics topics = topicsDAO.getRow(id);

if (products == null)

{

return HttpNotFound();

}

return View(topics);

}

// POST: Admin/Category/Delete/5:Xoa mot mau tin ra khoi CSDL

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public ActionResult DeleteConfirmed(int id)

{

Topics topics = topicsDAO.getRow(id);

//tim thay mau tin thi xoa, cap nhat cho Links

if (topicsDAO.Delete(topics) == 1)

{

Links links = linksDAO.getRow(topics.Id, "topic");

//Xoa luon cho Links

linksDAO.Delete(links);

}

//Thong bao thanh cong

TempData["message"] = new XMessage("success", "Xóa chủ đề thành công");

//O lai trang thung rac

return RedirectToAction("Trash");

}

### Chỉnh sửa tiêu đề trang xác nhận xóa dữ liệu (Delete)

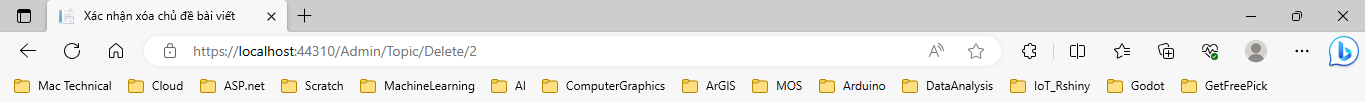
@model MyClass.Model.Topics

@{

ViewBag.Title = "Xác nhận xóa chủ đề bài viết";

Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}



### Chỉnh sửa menu chức năng (như Trash)

Lưu ý không thay thế nội dug vì trang này không sử dụng model mà giữ nguyên như phần tự động tạo ra như sau: @model MyClass.Model.Topics

<div class="card-header">

<div class="row">

<div class="col-md-6">

<strong class="text-danger">BẠN CÓ CHẮC CHẮN XÓA MỤC NÀY?</strong>

</div>

<div class="col-md-6 text-right">

<a href="@Url.Action("Index","Product")" class="btn-sm btn-info"> <i class="fa fa-long-arrow-alt-left"></i> Quay lại</a>

</div>

</div><!-- /.class-row -->

</div><!-- /.card-header -->

### Chỉnh sửa nội dung

Giữ nguyên phần Delete tự tạo bởi Entity, lư ý có form xác nhận

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<!-- Content Wrapper. Contains page content -->

…

}

#### Phần nội dung

@model MyClass.Model.Topics

@{

ViewBag.Title = "Xác nhận xóa chủ đề bài viết";

Layout = "~/Areas/Admin/Views/Shared/\_LayoutAdmin.cshtml";

}

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<!-- Content Wrapper. Contains page content -->

<div class="content-wrapper">

<!-- Main content -->

<section class="content my-2">

<!-- Default box -->

<div class="card">

<div class="card-header">

<div class="row">

<div class="col-md-6">

<strong class="text-danger">BẠN CÓ CHẮC CHẮN XÓA MỤC NÀY?</strong>

</div>

<div class="col-md-6 text-right">

<a href="@Url.Action("Index","Topic")" class="btn-sm btn-info"> <i class="fa fa-long-arrow-alt-left"></i> Quay lại</a>

</div>

</div>

</div>

<div class="card-body">

@\*Hien thi noi dung thong bao ngay ben tren bang du lieu\*@

@Html.Partial("Message")

<table class="table table-bordered table-striped" id="myTable">

<thead>

<tr>

<th>#</th>

<th>@Html.DisplayNameFor(model => model.Name)</th>

<th>@Html.DisplayNameFor(model => model.Slug)</th>

<th>@Html.DisplayNameFor(model => model.ParentId)</th>

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

<th>@Html.DisplayNameFor(model => model.Id)</th>

<th>Chức năng</th>

</tr>

</thead>

<tbody>

<tr>

<td><input type="checkbox" name="checkid" value="" /></td>

<td>@Html.DisplayFor(model => model.Name)</td>

<td>@Html.DisplayFor(model => model.Slug)</td>

<td>@Html.DisplayFor(model => model.ParentId)</td>

<td>@Html.DisplayFor(model => model.Order)</td>

<td>@Html.DisplayFor(model => model.Id)</td>

<td>

<button class="btn-sm btn-danger" type="submit" name="Delete"> Xóa [Delete] <i class="fa fa-trash"></i> </button>

</td>

</tr>

</tbody>

</table>

</div><!-- /.card-body -->

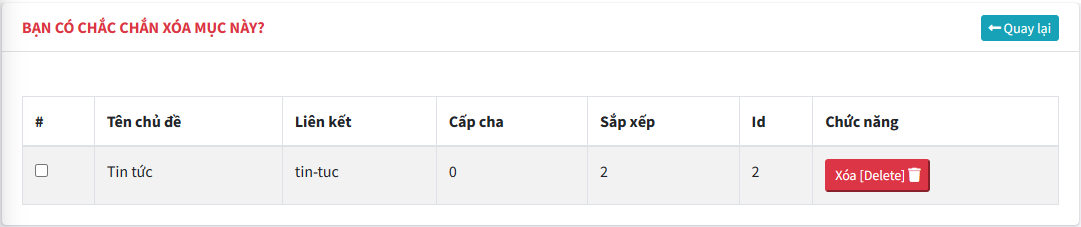
</div><!-- /.card -->

</section><!-- /.content -->

</div><!-- /.content-wrapper -->

}

## Trước khi xóa



## Sau khi xóa

