Phần 17: Kết hợp sắp xếp, tìm kiếm và phân trang

Sau khi học các chức năng sắp xếp, tìm kiếm và phân trang ở các bài viết trước, bạn sẽ học cách gom các chức năng này lại trong ASP.NET [MVC](https://www.dammio.com/glossary/mvc). Cách đơn giản nhất là gom hết code lại là dự án web của bạn có thể hoạt động theo ý muốn, nhưng bạn lưu ý đây chỉ là cách nghiệp dư. Một số lập trình viên lâu năm thường viết các đoạn code này ở 1 thư viện (hay 1 lớp nào đó) và khi cần chỉ nhúng vài dòng code là xong. Vì bạn là người mới học ASP.NET [MVC](https://www.dammio.com/glossary/mvc), cho nên [bài này](https://www.dammio.com/2018/11/21/asp-net-mvc-phan-17-ket-hop-sap-xep-tim-kiem-va-phan-trang) sẽ trình bày cách gom code lại trước.

Tiếp tục sử dụng dự án Web ở bài 16, bạn chỉnh sửa 1 số tập tin như sau.

# Sửa tập tin LinkController.cs

Bạn mở tập tin này và sửa như đoạn mã sau. Phần ba chấm (…) nghĩa là các code khác sau đó giữ nguyên.

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

using DammioMVC1.Models;

using System.ComponentModel;

using System.Linq.Dynamic;

using System.Linq.Expressions;

using System.Reflection;

using PagedList;

namespace DammioMVC1.Controllers

{

    public class LinkController : Controller

    {

        private DammioEntities db = new DammioEntities();

        public class HttpParamActionAttribute : ActionNameSelectorAttribute

        {

            public override bool IsValidName(ControllerContext controllerContext, string actionName, MethodInfo methodInfo)

            {

                if (actionName.Equals(methodInfo.Name, StringComparison.InvariantCultureIgnoreCase))

                    return true;

                var request = controllerContext.RequestContext.HttpContext.Request;

                return request[methodInfo.Name] != null;

            }

        }

        [HttpGet]

        // GET: /Link/

        public ActionResult Index(int? size, int? page, string sortProperty, string sortOrder, string searchString)

        {

            // 1. Tạo biến ViewBag gồm sortOrder, searchValue, sortProperty và page

            if (sortOrder == "asc") ViewBag.sortOrder = "desc";

            if (sortOrder == "desc") ViewBag.sortOrder = "";

            if (sortOrder == "") ViewBag.sortOrder = "asc";

            ViewBag.searchValue = searchString;

            ViewBag.sortProperty = sortProperty;

            ViewBag.page = page;

            // 2. Tạo danh sách chọn số trang

            List<SelectListItem> items = new List<SelectListItem>();

            items.Add(new SelectListItem { Text = "5", Value = "5" });

            items.Add(new SelectListItem { Text = "10", Value = "10" });

            items.Add(new SelectListItem { Text = "20", Value = "20" });

            items.Add(new SelectListItem { Text = "25", Value = "25" });

            items.Add(new SelectListItem { Text = "50", Value = "50" });

            items.Add(new SelectListItem { Text = "100", Value = "100" });

            items.Add(new SelectListItem { Text = "200", Value = "200" });

            // 2.1. Thiết lập số trang đang chọn vào danh sách List<SelectListItem> items

            foreach(var item in items)

            {

                if (item.Value == size.ToString()) item.Selected = true;

            }

            ViewBag.size = items;

            ViewBag.currentSize = size;

            // 3. Lấy tất cả tên thuộc tính của lớp Link (LinkID, LinkName, LinkURL,...)

            var properties = typeof(Link).GetProperties();

            List<Tuple<string, bool, int>> list = new List<Tuple<string, bool, int>>();

            foreach (var item in properties)

            {

                int order = 999;

                var isVirtual = item.GetAccessors()[0].IsVirtual;

                if (item.Name == "LinkName") order = 2;

                if (item.Name == "LinkID") order = 1;

                if (item.Name == "LinkDescription") order = 3;

                if (item.Name == "LinkURL") order = 4;

                if (item.Name == "CategoryID") continue; // Không hiển thị cột này

                Tuple<string, bool, int> t = new Tuple<string, bool, int>(item.Name, isVirtual, order);

                list.Add(t);

            }

            list = list.OrderBy(x => x.Item3).ToList();

            // 3.1. Tạo Heading sắp xếp cho các cột

            foreach (var item in list)

            {

                if (!item.Item2)

                {

                    if (sortOrder == "desc" && sortProperty == item.Item1)

                    {

                        ViewBag.Headings += "<th><a href='?page=" + page + "&size=" + ViewBag.currentSize + "&sortProperty=" + item.Item1 + "&sortOrder=" +

                            ViewBag.sortOrder + "&searchString=" + searchString + "'>" + item.Item1 + "<i class='fa fa-fw fa-sort-desc'></i></th></a></th>";

                    }

                    else if (sortOrder == "asc" && sortProperty == item.Item1)

                    {

                        ViewBag.Headings += "<th><a href='?page=" + page + "&size=" + ViewBag.currentSize + "&sortProperty=" + item.Item1 + "&sortOrder=" +

                            ViewBag.sortOrder + "&searchString=" + searchString + "'>" + item.Item1 + "<i class='fa fa-fw fa-sort-asc'></a></th>";

                    }

                    else

                    {

                        ViewBag.Headings += "<th><a href='?page=" + page + "&size=" + ViewBag.currentSize + "&sortProperty=" + item.Item1 + "&sortOrder=" +

                           ViewBag.sortOrder + "&searchString=" + searchString + "'>" + item.Item1 + "<i class='fa fa-fw fa-sort'></a></th>";

                    }

                }

                else ViewBag.Headings += "<th>" + item.Item1 + "</th>";

            }

            // 4. Truy vấn lấy tất cả đường dẫn

            var links = from l in db.Links

                        select l;

            // 5. Tạo thuộc tính sắp xếp mặc định là "LinkID"

            if (String.IsNullOrEmpty(sortProperty)) sortProperty = "LinkID";

            // 5. Sắp xếp tăng/giảm bằng phương thức OrderBy sử dụng trong thư viện Dynamic LINQ

            if (sortOrder == "desc") links = links.OrderBy(sortProperty + " desc");

            else if (sortOrder == "asc") links = links.OrderBy(sortProperty);

            else links = links.OrderBy("LinkID");

            // 5.1. Thêm phần tìm kiếm

            if (!String.IsNullOrEmpty(searchString))

            {

                links = links.Where(s => s.LinkName.Contains(searchString));

            }

            // 5.2. Nếu page = null thì đặt lại là 1.

            page = page ?? 1; //if (page == null) page = 1;

            // 5.3. Tạo kích thước trang (pageSize), mặc định là 5.

            int pageSize = (size ?? 5);

            ViewBag.pageSize = pageSize;

            // 6. Toán tử ?? trong C# mô tả nếu page khác null thì lấy giá trị page, còn

            // nếu page = null thì lấy giá trị 1 cho biến pageNumber. --- dammio.com

            int pageNumber = (page ?? 1);

            // 6.2 Lấy tổng số record chia cho kích thước để biết bao nhiêu trang

            int checkTotal = (int) (links.ToList().Count / pageSize) + 1;

            // Nếu trang vượt qua tổng số trang thì thiết lập là 1 hoặc tổng số trang

            if (pageNumber > checkTotal) pageNumber = checkTotal;

            // 7. Trả về các Link được phân trang theo kích thước và số trang.

            return View(links.ToPagedList(pageNumber, pageSize));

        }

        [HttpPost, HttpParamAction]

        public ActionResult Reset()

        {

            ViewBag.searchValue = "";

            Index(null, null, "", "", "");

            return View();

        }

// các code khác giữ nguyên

# Chỉnh sửa tập tin Views/Link/Index.cshtml

Tương tự, bạn cũng sửa tập tin View Index.cshtml như sau.

<!-- Thêm thư viện -->

@model PagedList.IPagedList<DammioMVC1.Models.Link>

@using PagedList.Mvc;

<!-- Kết thúc-->

@{

    ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

    @Html.ActionLink("Create New", "Create")

</p>

@using (Html.BeginForm("Index", "Link", FormMethod.Get))

{

    <p>

        LinkName: @Html.TextBox("SearchString", (string)ViewBag.searchValue) <input type="submit" value="Tìm kiếm" />

<input type="submit" name="Reset" value="Mặc định" />

        @Html.Hidden("page", (object)ViewBag.page)

        @Html.Hidden("sortProperty", (object)ViewBag.sortProperty)

        @Html.Hidden("sortOrder", (object)ViewBag.sortOrder)

        @Html.Hidden("size", (object)ViewBag.pageSize)

    </p>

}

<table class="table">

    <tr>

        @Html.Raw(ViewBag.Headings)

    </tr>

@foreach (var item in Model) {

    <tr>

        <td>

            @Html.DisplayFor(modelItem => item.LinkID)

        </td>

        <td>

            @Html.DisplayFor(modelItem => item.LinkName)

        </td>

        <td>

            @Html.DisplayFor(modelItem => item.LinkDescription)

        </td>

        <td>

            @Html.DisplayFor(modelItem => item.LinkURL)

        </td>

        <td>

            @Html.DisplayFor(modelItem => item.Category.CategoryName)

        </td>

        <td>

            @Html.ActionLink("Edit", "Edit", new { id=item.LinkID }) |

            @Html.ActionLink("Details", "Details", new { id=item.LinkID }) |

            @Html.ActionLink("Delete", "Delete", new { id=item.LinkID })

        </td>

    </tr>

}

</table>

<!-- Thêm mã phân trang -->

<br />

<div class="row">

    <div class="col-md-6 col-lg-6 col-xs-6 col-sm-6">

        @using (Html.BeginForm("Index","Link",FormMethod.Get))

        {

            <p>

                Kích thước trang: @Html.DropDownList("size", (List<SelectListItem>)ViewBag.size, new { @onchange = "this.form.submit();" })

            </p>

            @Html.Hidden("page", (object) ViewBag.page)

            @Html.Hidden("sortProperty", (object)ViewBag.sortProperty)

            @Html.Hidden("sortOrder", (object)ViewBag.sortOrder)

            @Html.Hidden("searchString", (object)ViewBag.searchValue)

        }

    </div>

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

        Trang: @(Model.PageCount < Model.PageNumber ? 0 : Model.PageNumber) / @Model.PageCount

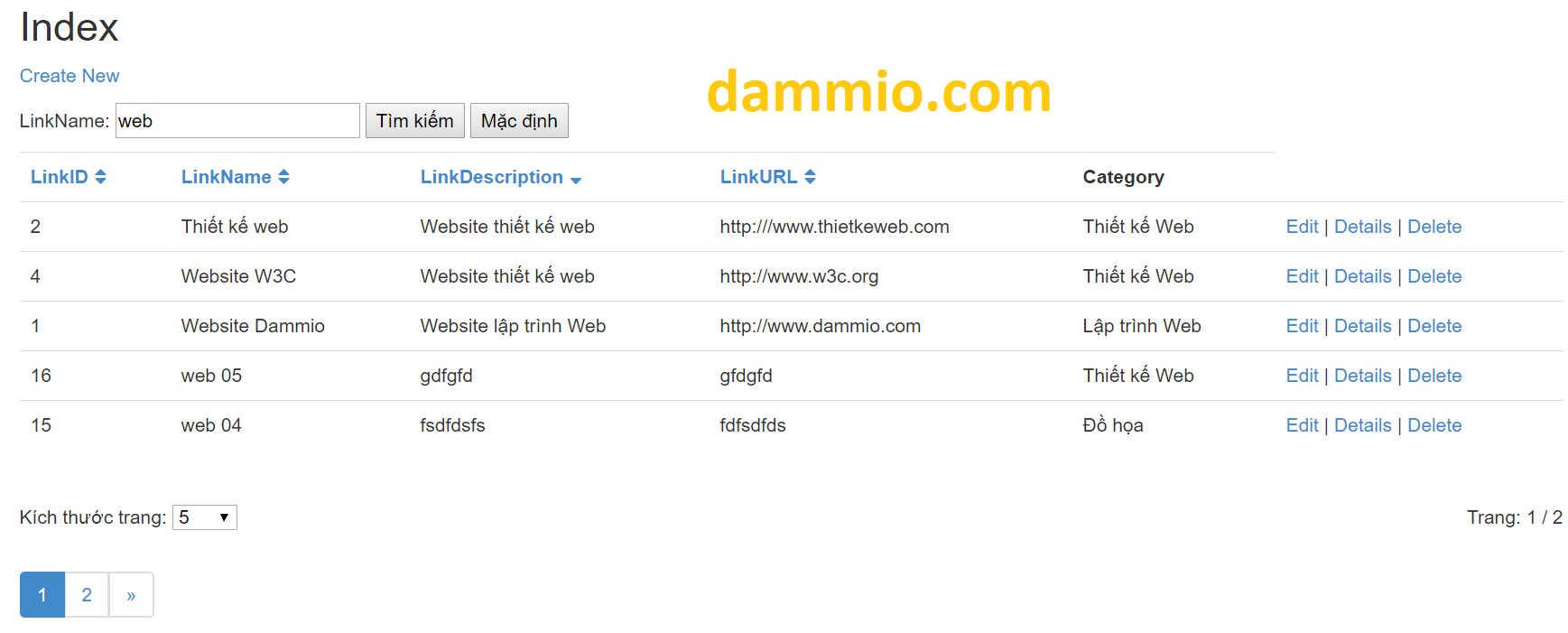
    </div>

</div>

@Html.PagedListPager(Model, page => Url.Action("Index", new { page, size = ViewBag.currentSize, sortProperty = ViewBag.sortProperty, sortOrder = ViewBag.SortOrder, searchString = ViewBag.searchValue }))

<!-- Kết thúc -->

Sau đó, bạn build dự án và chạy đường dẫn **http://localhost:45033/Link** để xem kết quả, thử nghiệm 1 số chức năng để kiểm tra xem hoạt động đúng hay không.



Lưu ý hãy nhập thật nhiều dữ liệu vào bảng Link để test hiệu quả hơn.

# Kết luận

Phần này chỉ hướng dẫn bạn gom code lại để thực hiện chức năng phân trang, tìm kiếm, sắp xếp đáp ứng nhu cầu cơ bản để quản lý 1 bảng cơ sở dữ liệu. Về cách tối ưu mã nguồn, bạn nên là người sáng tạo ra cách thức của riêng mình. Mời bạn tiếp tục theo dõi phần tiếp theo.