Phần 16: Phân trang trong ASP.NET MVC (mở rộng)

Trong phần này, bạn tiếp tục học cách thêm 1 tính năng phân trang với kích thước trang cho phép người dùng tự chọn trong danh sách DropDown có sẵn.

# Sửa phương thức Index trong tập tin Controller/ LinkController.cs

Bạn mở tập tin này và chỉnh sửa phương thức Index theo đoạn mã dưới đây. Trước hết, bạn thêm 2 tham số **int? size, int? page** vào phương thức và thuộc tính [HttpGet] được thêm ở đầu phương thức Index dùng để định nghĩa việc submit form ở View là kiểu HttpGet nhằm lấy biến dữ liệu trên [URL](https://www.dammio.com/glossary/url) (ví dụ: **http://localhost:xxxx/Link?page=2&size=10**). Chúng ta sử dụng một List với mỗi item kiểu SelectListItem để thêm giá trị kích thước trang. Mục 1.1 chúng ta thêm phần tử kích thước trang hiện thời đang được chọn. Mục 1.2 thì sử dụng 2 biến ViewBag, size (danh sách DropDown) và currentSize (kích thước trang hiện thời) để giữ trạng thái trên [URL](https://www.dammio.com/glossary/url).

[HttpGet]

// GET: /Link/

public ActionResult Index(int? size, int? page)

{

    // 1. Tạo list pageSize để người dùng có thể chọn xem để phân trang

    // Bạn có thể thêm bớt tùy ý --- dammio.com

    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" });

    // 1.1. Giữ trạng thái kích thước trang được chọn trên DropDownList

    foreach(var item in items)

    {

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

    }

    // 1.2. Tạo các biến ViewBag

    ViewBag.size = items; // ViewBag DropDownList

    ViewBag.currentSize = size; // tạo biến kích thước trang hiện tại

    // 2. Nếu page = null thì đặt lại là 1.

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

    // 3. Tạo truy vấn, lưu ý phải sắp xếp theo trường nào đó, ví dụ OrderBy

    // theo LinkID mới có thể phân trang.

    var links = (from l in db.Links

                 select l).OrderBy(x => x.LinkID);

    // 4. Tạo kích thước trang (pageSize), mặc định là 5.

    int pageSize = (size ?? 5);

    // 4.1 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.

    int pageNumber = (page ?? 1);

    // 5. Trả về các Link được phân trang theo kích thước và số trang.

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

}

Các mục khác như 3, 4, 4.1, và 5 không có gì thay đổi so với phần trước.

# Sửa giao diện Views/Link/Index.cshtml

Ở trang View Index.cshtml, chúng ta chủ yếu sửa ở phần cuối trang theo mã 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>

<table class="table">

    <tr>

        <th>

            LinkID

        </th>

        <th>

           LinkName

        </th>

        <th>

            LinkDescription

        </th>

        <th>

           LinkURL

        </th>

        <th>

           CategoryName

        </th>

    </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>

        }

    </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 }))

<!-- Kết thúc -->

Nhìn vào phần cuối trang, bạn có thể thấy chúng ta sử dụng 1 form với kiểu GET và submit về phương thức Index ở controller Link (LinkController.cs). Trong form này, chúng ta dùng 1 DropDownList lấy dữ liệu từ biến ViewBag.size và sử dụng sự kiện **onchange** để submit trực tiếp form mà không cần nút submit.

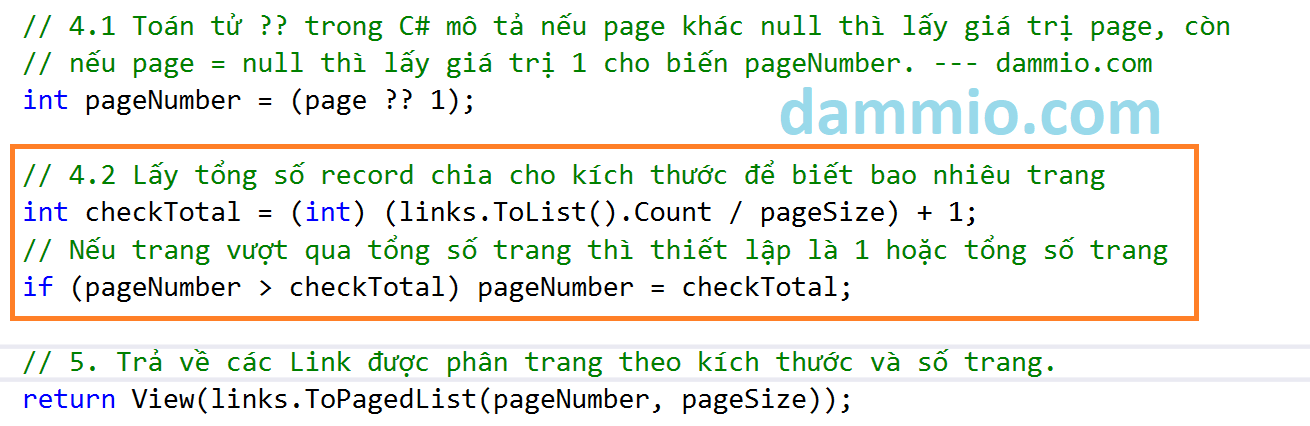
Ở phần phân trang, **[Url](https://www.dammio.com/glossary/url" \t "_blank).Action** là các liên kết phân trang với 2 tham số là page và size với size là kích thước trang hiện thời (mặc định size là 5).

# Thử nghiệm

Bạn build dự án, chạy đường dẫn http://localhost:xxxx/Link để xem kết quả như sau.  


Kích thước trang mặc định là 5 (size = 5), do đó bạn có cố tình hack injection kiểu **http://localhost:45033/Link?size=aaaaa** thì kết quả vẫn hiển thị mặc định, tức 5 Link trên 1 trang.

Trường hợp người dùng cố tình tạo số trang lớn (kiểu **http://localhost:45033/Link?page=999999**) thì đơn giản trang sẽ không hiện kết quả. Bạn có thể dùng mệnh đề if bắt lỗi ở phương thức Index nếu muốn như sau.



# Kết luận

Bài viết đã chỉ thêm cho bạn 1 phương pháp để phân trang mà kích thước trang người dùng có thể tự chọn. [DAMMIO](https://www.dammio.com/2018/11/02/asp-net-mvc-phan-16-phan-trang-trong-asp-net-mvc-mo-rong) mời bạn theo dõi phần tiếp theo nhé.