# 1:添加分页类 Pagination.cs

public class Pagination

{

public int? PageIndex { get; private set; }//当前页号减一的值

public int PageSize { get; private set; }　//每页显示的内容数量

public int TotalPages { get; private set; }//总页数

public int Start { get; private set; }//当前页面，显示的第一个页号（比如在中间的页面，页号显示是9号到16号，9就是Start）

public int End { get; private set; }//当前页面，显示的最后一个页号

// 为“上一页”“下一页”导航备用

public bool HasPreviousPage

{

get { return (PageIndex > 0); }

}

public bool HasNextPage

{

get { return (PageIndex + 1 < TotalPages); }

}

public Pagination(int? pageIndex, int pageSize, int totalcount)

{

PageIndex = (pageIndex ?? 0);

PageSize = pageSize; ;

TotalPages = (int)Math.Ceiling(totalcount / (double)PageSize);

int PaginationSize = 10;//这里规定每个页面显示10个页号

int size;//判定每个页面显示多少个页号

if (TotalPages > PaginationSize)

{

size = PaginationSize;

//定义每个页面的页号从几开始

if (pageIndex > 2 && pageIndex < TotalPages - (size - 2))

{

Start = (pageIndex ?? 0) - 1;

}

else if (pageIndex >= TotalPages - (size - 2))

{

Start = TotalPages - size + 1;

}

else

{

Start = 1;

}

}

else

{

size = TotalPages;

Start = 1;

}

End = Start + size - 1;

}

}

# 2:添加Htmlhelp扩展方法 ，作用是对URL进行操作和处理

public static class HtmlHelpers

{

public static string UrlSet(this HtmlHelper helper, HttpRequestBase request, string queryName, object queryValue)

{

if (queryName != null && queryValue != null)

{

string path = request.Path;

NameValueCollection query = new NameValueCollection(request.QueryString);

query.Set(queryName, queryValue.ToString());

return CreateUrl(path, query);

}

return request.RawUrl;

}

/// <summary>

///

/// </summary>

/// <param name="helper"></param>

/// <param name="request"></param>

/// <param name="queryName"></param>

/// <param name="queryValue"></param>

/// <returns></returns>

public static string UrlAdd(this HtmlHelper helper, HttpRequestBase request, string queryName, object queryValue)

{

if (queryName != null && queryValue != null)

{

string path = request.Path;

NameValueCollection query = new NameValueCollection(request.QueryString);

query.Add(queryName, queryValue.ToString());

return CreateUrl(path, query);

}

return request.RawUrl;

}

public static string UrlRemove(this HtmlHelper helper, HttpRequestBase request, string queryName)

{

if (queryName != null)

{

string path = request.Path;

NameValueCollection query = new NameValueCollection(request.QueryString);

query.Remove(queryName);

return CreateUrl(path, query);

}

return request.RawUrl;

}

public static string UrlRemove(this HtmlHelper helper, HttpRequestBase request, string queryName, object queryValue)

{

if (queryName != null && queryValue != null)

{

string path = request.Path;

NameValueCollection query = new NameValueCollection(request.QueryString);

string[] values = query.GetValues(queryName);

List<string> vs = values.ToList();

vs.Remove(queryValue.ToString());

query.Remove(queryName);

foreach (var item in vs)

{

query.Add(queryName, item);

}

return CreateUrl(path, query);

}

return request.RawUrl;

}

private static string CreateUrl(string path, NameValueCollection query)

{

string url = path;

if (query != null && query.Count != 0)

{

url += "?";

for (int i = 0; i < query.Count; i++)

{

string key = query.GetKey(i);

string[] values = query.GetValues(key);

for (int j = 0; j < values.Length; j++)

{

url += key + "=" + values[j] + "&";

}

}

url = url.TrimEnd('&');

}

return url;

}

}

# 3添加分页的分布视图页面\_Pagination.cshtml

@using BiosealMVCWeb.Models

@model BiosealMVCWeb.Models.Pagination

@{

ViewBag.Title = "\_Pagination";

}

<style type="text/css">

.current {

color: red;

text-decoration: none;

}

</style>

<div class="clearfix pagination">

<div class="pagination-inner">

@if (Model.Start > 1)

{

//如果当前页面超过第二页则显示第一页“1...”的链接

string url = Html.UrlSet(Request, "pageIndex", 0);

<a title="第一页1..." href="@url">1...</a>

}

@for (int i = Model.Start; i <= Model.End; i++)

{

string url = Html.UrlSet(Request, "pageIndex", i - 1);

if (i == Model.PageIndex + 1)

{

//在当前页面的几个页号中寻找当前页号

<a title="当前第 @i.ToString() 页" class="current" href="@url"

style="color: red;text-decoration: none; font-size:large"> @i.ToString() </a>

}

else

{

<a title="前往第 @i.ToString() " href="@url">@i.ToString()</a>

}

}

@if (Model.End < Model.TotalPages && Model.PageIndex - Model.Start < 2)

{

//最后一页

string totalPages = (@Model.TotalPages).ToString();

string url = Html.UrlSet(Request, "pageIndex", Model.TotalPages - 1);

<a title="最后一页 @totalPages" href="@url">... @totalPages</a>

}

总共 @Model.End 页 -- 当前第 @(Model.PageIndex+1) 页

</div>

</div>

3:添加分页视图\_Pagination.cshtml

@using BiosealMVCWeb.Models

@model BiosealMVCWeb.Models.Pagination

@{

ViewBag.Title = "\_Pagination";

}

<div class="pager">

<ul>

@if (Model.HasPreviousPage)

{

<li><a class="prev" href="@Html.UrlSet(Request, "pageIndex", (Model.PageIndex -1))">上一页</a></li>

}

@for (int i = Model.Start; i <= Model.End; i++)

{

string url = Html.UrlSet(Request, "pageIndex", i - 1);

if (i == Model.PageIndex + 1)

{

//在当前页面的几个页号中寻找当前页号

<li class="cur">

<a title="当前第 @i.ToString() 页" href="@url"> @i.ToString()</a>

</li>

}

else

{

<li>

<a title="第 @i.ToString() 页" href="@url"> @i.ToString()</a>

</li>

}

}

@if (Model.HasNextPage)

{

<li><a class="next" href="@Html.UrlSet(Request, "pageIndex", (Model.PageIndex +1))">下一页</a></li>

}

<li>共有 @Model.TotalCount 条记录</li>

</ul>

</div>

# 4 具体使用页面和控制器示例

## 页面使用分页视图

@Html.Partial("\_Pagination", (Linkmed\_inquiry.Models.Pagination)ViewBag.Pagination)

//传入Pagintion类

## 控制器中使用

设置页面参数

Linq表达式Skip和Take的使用

public ActionResult Campus()

{

var result = db.Campus;

int pageIndex = (Request.QueryString["pageIndex"]).QueryStringIntHelp();

int pageSize = 10; //设置每页显示条数

ViewBag.Pagination = new Pagination(pageIndex, pageSize, result.Count());

return View(result.OrderByDescending(i => i.Date).Skip(pageIndex \* pageSize).Take(pageSize).ToList());

}