CODER
FOUNDRY

Full Text Search (Blog)

This document will guide you through modifying your code to allow searching and filtering your list of Blogs in the Index of your project.

Find in your BlogPosts Controller, the "Get" `ActionResult` for Index. At this point you should have implemented paging for your application, and your Index method may look something like this.

```
public ActionResult Index(int? page)
{
    pageSize = 5;  // the number of posts you want to display per page
    int pageNumber = (page ?? 1);

    var listPosts = db.BlogPosts.AsQueryable();
    return View(listPosts.OrderByDescending(p => p.Created).ToPagedList(pageNumber, pageSize));
}
```

Modify your Index() code to look like this…

```
        public ActionResult Index(int? page, string searchStr)
        {
            ViewBag.Search = searchStr;
            var blogList = IndexSearch(searchStr);

            int pageSize = 5; // the number of posts you want to display per page
            int pageNumber = (page ?? 1);

            return View(blogList.ToPagedList(pageNumber, pageSize));
        }
```

If you have a **Post method** for the ActionResult Index(), **Delete it.**

Add the following method.

```
    public IQueryable<BlogPost> IndexSearch(string searchStr)
     {
         IQueryable<BlogPost> result = null;
         if (searchStr != null)
         {
             result = db.Posts.AsQueryable();
             result = result.Where(p => p.Title.Contains(searchStr) ||
                                  p.Body.Contains(searchStr) ||
                                  p.Comments.Any(c => c.Body.Contains(searchStr) ||
                                                 c.Author.FirstName.Contains(searchStr) ||
                                                 c.Author.LastName.Contains(searchStr) ||
                                                 c.Author.DisplayName.Contains(searchStr) ||
                                                 c.Author.Email.Contains(searchStr)));
         }
         else
         {
             result = db.Posts.AsQueryable();
         }

         return result.OrderByDescending(p => p.Created);
     }
```

Now in your Index View **Add or Update the search form with the following code**. Pay close attention to the BeginForm parameter list!

```
@using (Html.BeginForm("Index", "BlogPosts", FormMethod.Get, htmlAttributes: new { @class=  "form" }))
{
   <div class="controls controls-row search-row">
      <input id="searchStr" name="searchStr" class="input-block-level" type="text"
                  placeholder="Search">
      <button id="searchStr" name="searchStr" class="btn btn-success" type="submit">
         <i class="icon-search"></i>
      </button>
   </div>
}
```

Lastly, **Add or Modify your Paging controls with the following code**.

```
@Html.PagedListPager(Model, page => Url.Action("Index", new {page, searchStr=ViewBag.Search
   }));
```