عنوان: **یافتن «مطالب مرتبط» توسط لوسین**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۳۱ ۱۷:۲۵
آدرس: www.dotnettips.info
گروه‌ها: Lucene.NET

موتور لوسین علاوه بر فراهم آوردن امکان جستجوی سریع بر روی متون حجیم، امکان یافتن مطالبی مشابه یا مرتبط با مطلبی خاص را نیز فراهم می‌کند. نمونه آن‌را شاید در بعضی از انجمن‌ها یا وبلاگ‌ها دیده باشید که در ذیل مطلب جاری، چندین لینک را به مطالبی مشابه نیز نمایش می‌دهند. در ادامه نحوه استفاده از این قابلیت را در لوسین بررسی خواهیم کرد.

یافتن شماره سند متناظر لوسین

همان مثال « استفاده از لوسین برای برجسته سازی عبارت جستجو شده در نتایج حاصل » را در نظر بگیرید. در ابتدا نیاز است شماره یک مطلب را تبدیل به شماره سند لوسین کنیم. برای مثال ممکن است Id یک مطلب 1000 باشد، اما شماره سند متناظر آن در لوسین 800 ثبت شده باشد. بنابراین جستجوی ذیل الزامی است:

```
static readonly Lucene.Net.Util.Version _version = Lucene.Net.Util.Version.LUCENE_29;
static readonly IndexSearcher _searcher = new IndexSearcher(@"c:\path\idx", readOnly: true);
private static int GetLuceneDocumentNumber(int postId)
{
    var analyzer = new StandardAnalyzer(_version);
    var parser = new QueryParser(_version, "Id", analyzer);
    var query = parser.Parse(postId.ToString());
    var doc = _searcher.Search(query, 1);
    if (doc.totalHits == 0)
    {
        return 0;
    }
    return doc.scoreDocs[0].doc;
}
```

در اینجا بر اساس شماره یک مطلب، کوئری متناظر با آن تشکیل شده و جستجویی بر روی اسناد ثبت شده در ایندکس‌های لوسین صورت می‌گیرد. اگر اطلاعاتی یافت شد، شماره سند متناظر بازگشت داده می‌شود.
از این جهت به شماره سند یاد شده نیاز داریم که قرار است مطالب مرتبط با کل این سند را بیابیم.

ساختن کوئری‌های MoreLikeThis

امکانات یافتن مطالب مشابه یک مطلب در اسمبلی Lucene.Net.Contrib.Queries.dll قرار دارد. بنابراین در اینجا نیاز به فایل‌های پروژه Lucene.Net Contrib وجود دارد.
پس از یافتن شماره سند متناظر با یک مطلب، اکنون نوبت به ساخت کوئری‌های پیشرفته MoreLikeThis است که نحوه انجام تنظیمات آن‌را در ذیل مشاهده می‌کنید:

```
private static Query CreateMoreLikeThisQuery(int postId)
{
    var docNum = GetLuceneDocumentNumber(postId);
    if (docNum == 0)
            return null;

     var analyzer = new StandardAnalyzer(_version);
     var reader = _searcher.GetIndexReader();

     var moreLikeThis = new MoreLikeThis(reader);
     moreLikeThis.SetAnalyzer(analyzer);
     moreLikeThis.SetFieldNames(new[] { "Title", "Body"});
     moreLikeThis.SetMinDocFreq(1);
     moreLikeThis.SetMinTermFreq(1);
     moreLikeThis.SetBoost(true);

     return moreLikeThis.Like(docNum);
```

```
}
```

در اینجا فیلدهایی که قرار است در جستجو حضور داشته باشند توسط متد SetFieldNames معرفی می‌شوند. توسط متد SetMinDocFreq مشخص می‌کنیم که واژه‌های مشابه و مرتبط باید حداقل در چند سند ظاهر شده باشند. همچنین توسط متد SetMinTermFreq تعیین می‌گردد که یک واژه باید چندبار در این اسناد وجود داشته باشد. متد SetBoost سبب می‌شود که آنالیز بهتری بر اساس رتبه بندی‌های حاصل صورت گیرد.

نمایش مطالب مرتبط توسط کوئری MoreLikeThis

پس از این تنظیمات، متد moreLikeThis.Like، یک شیء Query را در اختیار ما قرار خواهد داد. از اینجای کار به بعد همانند سایر مطالب مشابه است. بر اساس این کوئری، جستجویی صورت گرفته و سپس اطلاعات یافت شده نمایش داده می‌شود:

```
private static void ShowMoreLikeThisPostItems(int postId)
{
     var query = CreateMoreLikeThisQuery(postId);
     if (query == null)
               return;

      var hits = _searcher.Search(query, n: 10);
      foreach (var item in hits.scoreDocs)
      {
           var doc = _searcher.Doc(item.doc);
           var id = doc.Get("Id");
           var title = doc.Get("Title");
           Console.WriteLine(title);
       }
}
```

نظرات خوانندگان

نویسنده: دل محسن
تاریخ: ۱۳۹۳/۰۷/۰۸ ۱۰:۴۸

سلام؛ میشه در Lucene جستجویی شبیه sql انجام داد؟ منظور اینکه همزمان item1 رو داخل field1 و item2 رو در field2 و.... جستجو کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۰۸ ۱۰:۵۳

بله. زبان مخصوص خودش را دارد: Query Parser Syntax و Lucene Query Syntax
ضمنا بر همین مبنا LINQ to Lucene هم طراحی شده: ^ و ^