

在 Apache Solr 中使用复制字段 (Copying Fields)

在 [《Apache Solr 安装部署及索引创建》](#) 文章里面我创建了一个名为 iteblog 的 core，并在里面导入了一些测试数据，然后在 [《使用 Apache Solr 检索数据》](#) 里面介绍了 Solr 中一些简单的查询。可能有同学按照上面文章介绍，在使用下面的查询发现啥都查不到：

```
[root@iteblog.com /opt/solr-7.4.0]$ curl http://iteblog.com:8983/solr/iteblog/select?q=electronics
{
  "responseHeader":{
    "status":0,
    "QTime":189,
    "params":{
      "q":"electronics"}},
  "response":{"numFound":0,"start":0,"docs":[]
}}
```

这是为什么呢？读了本文你就明白了。



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

使用 Solr 检索数据的时候，其实我们需要指定检索的字段，比如下面的查询：

```
[root@iteblog.com /opt/solr]$ curl http://iteblog.com:8983/solr/iteblog/select?q=manu:electronics
{
  "responseHeader":{
    "status":0,
    "QTime":4,
    "params":{
      "q":"manu:electronics"}},
  "response":{"numFound":1,"start":0,"docs":[
    {
      "id":"SP2514N",
      "name":"Samsung SpinPoint P120 SP2514N - hard drive - 250 GB - ATA-133",
      "manu":"Samsung Electronics Co. Ltd.",
      "manu_id_s":"samsung",
      "cat":["electronics",
        "hard drive"],
      "features":["7200RPM, 8MB cache, IDE Ultra ATA-133",
        "NoiseGuard, SilentSeek technology, Fluid Dynamic Bearing (FDB) motor"],
      "price":92.0,
      "price_c":"92.0,USD",
      "popularity":6,
      "inStock":true,
      "manufacturedate_dt":"2006-02-13T15:26:37Z",
      "store":"35.0752,-97.032",
      "_version_":1606843234767601664,
      "price_c____l_ns":9200}}
  ]}
}
```

我们指定了 manu:electronics，其含义是在 manu 字段里面检索出现 electronics 关键字的文档。但是当我们指定 q=electronics 的时候，Solr 内部默认是检索 _text_ 字段的，如下（参见 solrconfig.xml 配置文件）：

```
<initParams path="/update/**,/query,/select,/tvrh,/elevate,/spell,/browse">
  <lst name="defaults">
    <str name="df">_text_</str>
  </lst>
</initParams>
```

但问题是我们并没有为这个字段设定索引数据啊，所以这种情况下直接使用 `q=electronics` 去检索数据是无法查询到的。不过 Solr 为我们提供了 Copying Fields 机制，就是说我们可以将几个不同字段的数据拷贝到一个字段里面进行索引，那么具体咋实现呢？下面听我慢慢讲来。

假设我们有 `title`、`author`、`content`、`abstract` 等四个字段，在 `schema.xml` 里面我们可以定义如下：

```
<field name="title" type="text_gen_sort" indexed="true" />
<field name="author" type="text_gen_sort" indexed="true"/>
<field name="content" type="text_gen_sort" indexed="true"/>
<field name="abstract" type="text_gen_sort" indexed="true"/>
```

然后我们导完数据之后就可以通过 Solr 去搜索相关的数据，但是我们只能通过 `q=title:xxx`、`q=author:xxx`、`q=content:xxx` 以及 `q=abstract:xxx` 等方法去搜索。如果我们想使用 `q=xxx` 去搜索，那对不起，你搜索不到！这是因为如果你不指定搜索字段，Solr 默认将去名为 `_text_` 字段里面搜索，但这时候那个字段是没有任何数据的。如果我们想使用 `q=xxx` 搜索 `title`、`author`、`content`、`abstract` 字段里面的数据，那么你需要进行一些设置。

Solr 为我们提供了以下的方法：

```
<copyField source="xxx" dest="xxx" />
```

意思是将 `source` 字段里面的东西拷贝到 `dest` 字段里面去。聪明的同学肯定想到我们是不是可以将上面四个字段里面的内容拷贝到另外一个字段里面，然后就可以不指定搜索字段即可搜索全部字段里面的值呢？没错，确实是这样的。我们先定义一个字段，比如 `iteblog`，我们想把所有字段里面的数据拷贝到这个字段里面，可以定义如下：

```
<field name="iteblog" type="text_general" indexed="true" stored="false" multiValued="true"/>
```

注意 `multiValued="true"` 属性，有了它，`iteblog` 字段就可以存储多个字段的值，那么我们如何把其他字段里面的值拷贝到 `iteblog` 字段里面去呢？可以如下定义：

```
<copyField source="title" dest="iteblog" />
<copyField source="author" dest="iteblog" />
<copyField source="content" dest="iteblog" />
<copyField source="abstract" dest="iteblog" />
```

然后到 solrconfig.xml 文件里面设置如下：

```
<initParams path="/update/**,/query,/select,/tvrh,/elevate,/spell,/browse">  
  <lst name="defaults">  
    <str name="df">iteblog</str>  
  </lst>  
</initParams>
```

配置文件设置好之后，我们就可以创建 Core/Collection，然后导入数据建完索引，我们就可以通过 q=xxx 搜索 title、author、content、abstract 字段里面的数据。

使用 Solr 的 copyField 机制带来的问题也很明显，那就是它会增加我们索引数据的大小，所以在使用这个机制的时候，尽量想好哪些字段需要加入到默认搜索里面。不过 Solr 也为 copyField 提供了一个可以限制长度的参数：

```
<copyField source="cat" dest="text" maxChars="30000" />
```

maxChars 参数就是用于限制从 source 最多复制多少长度的内容到 dest 里面，其接收一个 int 类型的值。

在使用 copyField 属性的时候，我们还可以设置通配符，如下：

```
<copyField source="*_t" dest="text" maxChars="25000" />
```

上面把所有符合 *_t 规则的字段内容复制到 text 里面。

同时，一个字段可以到多个其他字段里面，如下：

```
<copyField source="here" dest="there"/>  
<copyField source="here" dest="elsewhere"/>
```

但是下面的写法就不支持：

```
<copyField source="here" dest="there"/>
<copyField source="there" dest="elsewhere"/>
```

上面关于 copyField 的用法是将其他字段里面的值拷贝到一个字段里面，然后我们就可以通过这个字段来搜索多个字段里面的值。copyField 的另一个用法就是对同一个字段使用多种分析方法，如下：

```
<field name="title" type="text"/>
<field name="text_gen_sort" type="text_gen_sort" stored="false"/>
<field name="text_exact" type="text_exact" stored="false"/>

<copyField source="title" dest="text_gen_sort"/>
<copyField source="title" dest="text_exact"/>
```

上面对 title 字段使用了三种方法去解析。

更多关于 Copying Fields 的使用，请参见官方文档：[Copying Fields](#)

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: **【】** ()