

Apache Solr 自定义数据模式

到目前为止，我们往 Solr 里面导数据都没有定义模式，也就是说让 Solr 去猜我们数据的类型以及解析方式，这种方式成为无模式（Schemaless）。Apache Solr 里面的定义为：One reason for this is we're going to use a feature in Solr called "field guessing", where Solr attempts to guess what type of data is in a field while it's indexing it. It also automatically creates new fields in the schema for new fields that appear in incoming documents. This mode is called "Schemaless".

。无模式虽然使用起来很方便，但是有时候会带来一些问题，正如下面的实例一样。

我们首先创建一个名为 films 的 core：

```
[root@iteblog.com /opt/solr]$ bin/solr create -c films -s 2 -rf 2
WARNING: Using _default configset with data driven schema functionality. NOT RECOMMENDED for production use.
```

```
To turn off: bin/solr config -c films -p 8983 -action set-user-property -property update.autoCreateFields -value false
INFO - 2018-07-25 11:43:53.888; org.apache.solr.util.configuration.SSLCredentialProviderFactory; Processing SSL Credential Provider chain: env;sysprop
```

Created new core 'films'

我们创建 films 的时候并没有指定任何的 configset，所以会使用 _default 里面的配置。现在我们来往里面导一些数据：

```
[root@iteblog.com /opt/solr]$ bin/post -c films example/films/films.json
/data/web/java/jdk1.8.0_181/bin/java -classpath /opt/solr/dist/solr-core-7.4.0.jar -Dauto=yes -Dc=films -Ddata=files org.apache.solr.util.SimplePostTool example/films/films.json
SimplePostTool version 5.0.0
Posting files to [base] url http://localhost:8983/solr/films/update...
Entering auto mode. File endings considered are xml,json,jsonl,csv,pdf,doc,docx,ppt,pptx,xls,xlsx,odt,odp,ods,ott,otp,ots,rtf,htm,html,txt,log
POSTing file films.json (application/json) to [base]/json/docs
SimplePostTool: WARNING: Solr returned an error #400 (Bad Request) for url: http://localhost:8983/solr/films/update/json/docs
SimplePostTool: WARNING: Response: {
  "responseHeader":{
    "status":400,
    "QTime":208},
```

```
"error":{
  "metadata":[
    "error-class","org.apache.solr.common.SolrException",
    "root-error-class","java.lang.NumberFormatException",
    "msg":"ERROR: [doc=/en/quien_es_el_senor_lopez] Error adding field 'name'=' ¿Qui ¿ n e
s el se ±or L ¿pez?' msg=For input string: W" ¿Qui ¿ n es el se ±or L ¿pez?W"',
    "code":400}}
```

SimplePostTool: WARNING: IOException while reading response: java.io.IOException: Server re
turned HTTP response code: 400 for URL: http://localhost:8983/solr/films/update/json/docs
1 files indexed.

COMMITting Solr index changes to http://localhost:8983/solr/films/update...

Time spent: 0:00:00.816

如果是 Windows , 可以使用下面命令导数 :

```
C:\solr-7.4.0> java -jar -Dc=films -Dauto example\Wexampledocs\Wpost.jar example\Wfilms\W*
.json
```

从上面的输出可以看出, 导数遇到了问题! 这是为什么呢? 因为我们使用的是无模式场景, 我们
试图让 Solr 帮我们推断数据的类型, 在上面的实例中, name
字段一开始被解析成数字类型的, 但是在解析后面的数据是发现又出现了非数字类型的行 (For
input string: W" ¿Qui ¿ n es el se ±or L ¿pez?W"), 所以导致数字转换异常 ("root-error-
class","java.lang.NumberFormatException"))。下面是 Solr 给我们猜测的字段类型 :

```
<field name="directed_by" type="text_general"/>
<field name="genre" type="text_general"/>
<field name="id" type="string" multiValued="false" indexed="true" required="true" stored="tr
ue"/>
<field name="initial_release_date" type="pdates"/>
<field name="name" type="pdoubles"/>
```

上面五个就是 Solr 帮我们解析出来的数据类型, 可以看出, name 这个字段被解析成 pdoubles
的类型。但实际上 name
这个字段其实是电影的名字, 只不过恰好前面几部电影的名字是数字, 导致 Solr
解析错误, 下面就是正常数据的内容

```
{
```

```
"id": "/en/quien_es_el_senor_lopez",
"directed_by": [
  "Luis Mandoki"
],
"genre": [
  "Documentary film"
],
"name": "¿Quié es el señor López?"
}
```

在这种情况下，我们需要对 name 字段自定义类型，可以使用下面的命令来修改 name 的类型：

```
[root@iteblog.com /opt/solr]$ curl -X POST -H 'Content-type:application/json' --data-binary '{"add-field": {"name":"name", "type":"text_general", "multiValued":false, "stored":true}}' http://iteblog.com:8983/solr/films/schema
{
  "responseHeader":{
    "status":0,
    "QTime":215}}
```

当然，我们也可以 Admin UI 界面修改，如下：



Dashboard

Logging

Cloud

Collections

Java Properties

Thread Dump

films

Overview

Analysis

Dataimport

Documents

Files

Query

Stream

Schema

Add Field
Add Dynamic Field
Add Copy Field

name:

field type:

default:

☒ stored
☐ indexed
☐ docValues
☐ multiValued
☐ required

Show omit options ☒
Show term vector options ☒
Show sort options ☒

Add Field
Cancel

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

修改完之后，我们可以看到 name 的配置已经发生了变化了，如下：

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

现在我们可以再一次导数了，这一次不会出错，如下：

```
[root@iteblog.com /opt/solr]$ bin/post -c films example/films/films.json
/data/web/java/jdk1.8.0_181/bin/java -classpath /opt/solr/dist/solr-core-7.4.0.jar -Dauto=yes -
Dc=films -Ddata=files org.apache.solr.util.SimplePostTool example/films/films.json
SimplePostTool version 5.0.0
Posting files to [base] url http://localhost:8983/solr/films/update...
Entering auto mode. File endings considered are xml,json,jsonl,csv,pdf,doc,docx,ppt,pptx,xls,xl
sx,odt,odp,ods,ott,otp,ots,rtf,htm,html,txt,log
```

```
POSTing file films.json (application/json) to [base]/json/docs
1 files indexed.
COMMITting Solr index changes to http://localhost:8983/solr/films/update...
Time spent: 0:00:01.783
```

当然，我们其实也可以在配置文件里面定义的，这里就不再介绍了。

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: [【】](#)（）