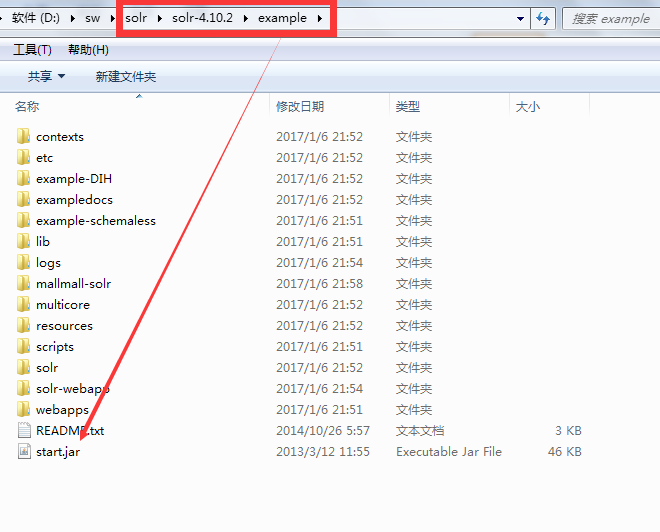
1. 下载solr

<http://lucene.apache.org/solr/downloads.html>

1. 解压



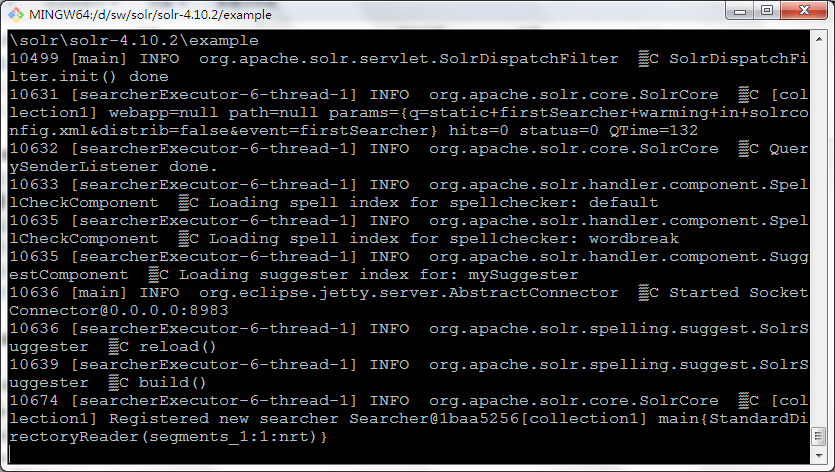
进入目录：



在目录下打开命令窗口

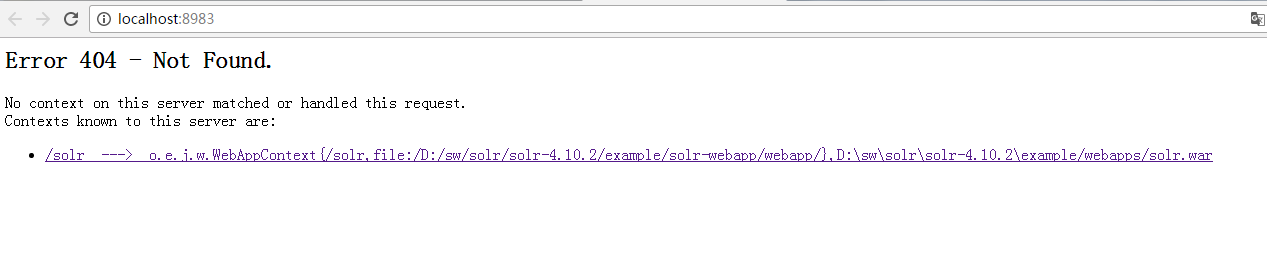
输入命令

Java -jar start.jar

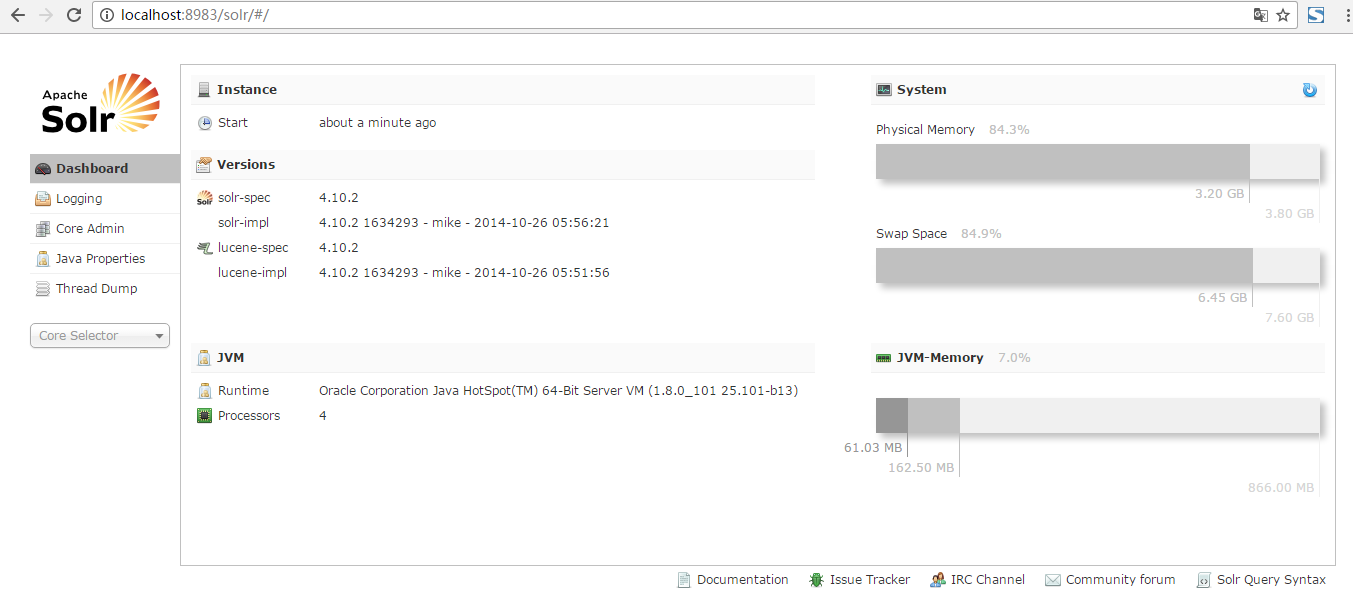


启动成功。

输入地址：<http://localhost:8983/>



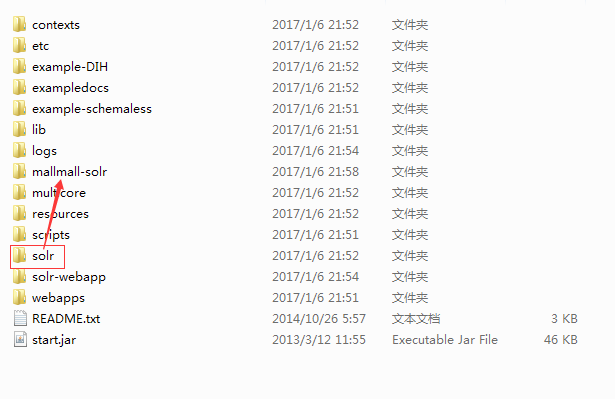
点击 /solr ---🡪 ……..



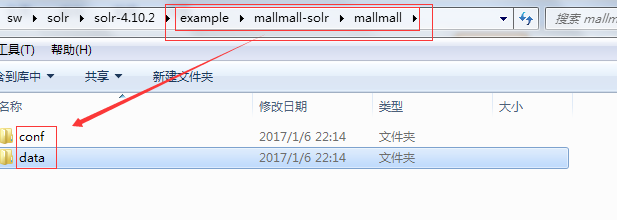
启动成功并进入管理后台。

创建 solr core

1. 进入 example 目录创建 mallmall-solr （自定义）
2. 将 solr 目录里的配置文件拷贝到自己的项目中



1. 进入自己的项目 mallmall-solr 创建目录 mallmall
2. 在 mallmall 目录创建 conf 和 data 目录



1. solr-4.10.2\example\solr\collection1\core.properties 文件拷贝到 example\mallmall-solr\mallmall 并且修改该配置文件里的配置 name=mallmall
2. example\solr\collection1\conf 里的配置文件 schema.xml、solrconfig.xml 拷贝到 example\mallmall-solr\mallmall\conf
3. 修改 schema.xml 配置文件，简化至如下：

<?xml version="1.0" encoding="UTF-8" ?>

<!--

Licensed to the Apache Software Foundation (ASF) under one or more

contributor license agreements. See the NOTICE file distributed with

this work for additional information regarding copyright ownership.

The ASF licenses this file to You under the Apache License, Version 2.0

(the "License"); you may not use this file except in compliance with

the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

-->

<!--

This is the Solr schema file. This file should be named "schema.xml" and

should be in the conf directory under the solr home

(i.e. ./solr/conf/schema.xml by default)

or located where the classloader for the Solr webapp can find it.

This example schema is the recommended starting point for users.

It should be kept correct and concise, usable out-of-the-box.

For more information, on how to customize this file, please see

http://wiki.apache.org/solr/SchemaXml

PERFORMANCE NOTE: this schema includes many optional features and should not

be used for benchmarking. To improve performance one could

- set stored="false" for all fields possible (esp large fields) when you

only need to search on the field but don't need to return the original

value.

- set indexed="false" if you don't need to search on the field, but only

return the field as a result of searching on other indexed fields.

- remove all unneeded copyField statements

- for best index size and searching performance, set "index" to false

for all general text fields, use copyField to copy them to the

catchall "text" field, and use that for searching.

- For maximum indexing performance, use the ConcurrentUpdateSolrServer

java client.

- Remember to run the JVM in server mode, and use a higher logging level

that avoids logging every request

-->

<schema name="example" version="1.5">

<!-- attribute "name" is the name of this schema and is only used for display purposes.

version="x.y" is Solr's version number for the schema syntax and

semantics. It should not normally be changed by applications.

1.0: multiValued attribute did not exist, all fields are multiValued

by nature

1.1: multiValued attribute introduced, false by default

1.2: omitTermFreqAndPositions attribute introduced, true by default

except for text fields.

1.3: removed optional field compress feature

1.4: autoGeneratePhraseQueries attribute introduced to drive QueryParser

behavior when a single string produces multiple tokens. Defaults

to off for version >= 1.4

1.5: omitNorms defaults to true for primitive field types

(int, float, boolean, string...)

-->

<!-- Valid attributes for fields:

name: mandatory - the name for the field

type: mandatory - the name of a field type from the

<types> fieldType section

indexed: true if this field should be indexed (searchable or sortable)

stored: true if this field should be retrievable

docValues: true if this field should have doc values. Doc values are

useful for faceting, grouping, sorting and function queries. Although not

required, doc values will make the index faster to load, more

NRT-friendly and more memory-efficient. They however come with some

limitations: they are currently only supported by StrField, UUIDField

and all Trie\*Fields, and depending on the field type, they might

require the field to be single-valued, be required or have a default

value (check the documentation of the field type you're interested in

for more information)

multiValued: true if this field may contain multiple values per document

omitNorms: (expert) set to true to omit the norms associated with

this field (this disables length normalization and index-time

boosting for the field, and saves some memory). Only full-text

fields or fields that need an index-time boost need norms.

Norms are omitted for primitive (non-analyzed) types by default.

termVectors: [false] set to true to store the term vector for a

given field.

When using MoreLikeThis, fields used for similarity should be

stored for best performance.

termPositions: Store position information with the term vector.

This will increase storage costs.

termOffsets: Store offset information with the term vector. This

will increase storage costs.

required: The field is required. It will throw an error if the

value does not exist

default: a value that should be used if no value is specified

when adding a document.

-->

<!-- field names should consist of alphanumeric or underscore characters only and

not start with a digit. This is not currently strictly enforced,

but other field names will not have first class support from all components

and back compatibility is not guaranteed. Names with both leading and

trailing underscores (e.g. \_version\_) are reserved.

-->

<!-- If you remove this field, you must \_also\_ disable the update log in solrconfig.xml

or Solr won't start. \_version\_ and update log are required for SolrCloud

-->

<field name="\_version\_" type="long" indexed="true" stored="true"/>

<!-- points to the root document of a block of nested documents. Required for nested

document support, may be removed otherwise

-->

<field name="\_root\_" type="string" indexed="true" stored="false"/>

<!-- Only remove the "id" field if you have a very good reason to. While not strictly

required, it is highly recommended. A <uniqueKey> is present in almost all Solr

installations. See the <uniqueKey> declaration below where <uniqueKey> is set to "id".

-->

<field name="id" type="string" indexed="true" stored="true" required="true" multiValued="false" />

<uniqueKey>id</uniqueKey>

<fieldType name="string" class="solr.StrField" sortMissingLast="true" />

<!-- boolean type: "true" or "false" -->

<fieldType name="boolean" class="solr.BoolField" sortMissingLast="true"/>

<!-- sortMissingLast and sortMissingFirst attributes are optional attributes are

currently supported on types that are sorted internally as strings

and on numeric types.

This includes "string","boolean", and, as of 3.5 (and 4.x),

int, float, long, date, double, including the "Trie" variants.

- If sortMissingLast="true", then a sort on this field will cause documents

without the field to come after documents with the field,

regardless of the requested sort order (asc or desc).

- If sortMissingFirst="true", then a sort on this field will cause documents

without the field to come before documents with the field,

regardless of the requested sort order.

- If sortMissingLast="false" and sortMissingFirst="false" (the default),

then default lucene sorting will be used which places docs without the

field first in an ascending sort and last in a descending sort.

-->

<!--

Default numeric field types. For faster range queries, consider the tint/tfloat/tlong/tdouble types.

These fields support doc values, but they require the field to be

single-valued and either be required or have a default value.

-->

<fieldType name="int" class="solr.TrieIntField" precisionStep="0" positionIncrementGap="0"/>

<fieldType name="float" class="solr.TrieFloatField" precisionStep="0" positionIncrementGap="0"/>

<fieldType name="long" class="solr.TrieLongField" precisionStep="0" positionIncrementGap="0"/>

<fieldType name="double" class="solr.TrieDoubleField" precisionStep="0" positionIncrementGap="0"/>

</schema>

1. 根据自己的表结构修改 schema.xml

<?xml version="1.0" encoding="UTF-8" ?>

<schema name="example" version="1.5">

<field name="\_version\_" type="long" indexed="true" stored="true"/>

<field name="\_root\_" type="string" indexed="true" stored="false"/>

<!--根据自己表结构修改字段 -->

<field name="id" type="long" indexed="true" stored="true" required="true" multiValued="false" />

<field name="title" type="string" indexed="true" stored="true" />

<field name="sellPoint" type="string" indexed="false" stored="true" />

<field name="price" type="long" indexed="true" stored="true" />

<field name="image" type="string" indexed="false" stored="true" />

<field name="cid" type="long" indexed="true" stored="true" />

<field name="status" type="int" indexed="true" stored="false" />

<uniqueKey>id</uniqueKey>

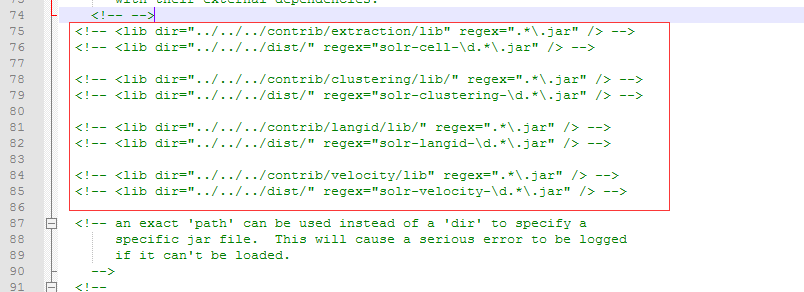
<fieldType name="string" class="solr.StrField" sortMissingLast="true" />

<fieldType name="int" class="solr.TrieIntField" precisionStep="0" positionIncrementGap="0"/>

<fieldType name="long" class="solr.TrieLongField" precisionStep="0" positionIncrementGap="0"/>

</schema>

1. 修改 solrconfig.xml
   1. 注释所有的 <lib> 标签



* 1. 替换默认搜索字段 <str name="df">text</str> 替换为<str name="df">title</str>
  2. 注释一下标签包含内容,起作用类似于竞价排名

<searchComponent name="elevator" class="solr.QueryElevationComponent" >

<!-- pick a fieldType to analyze queries -->

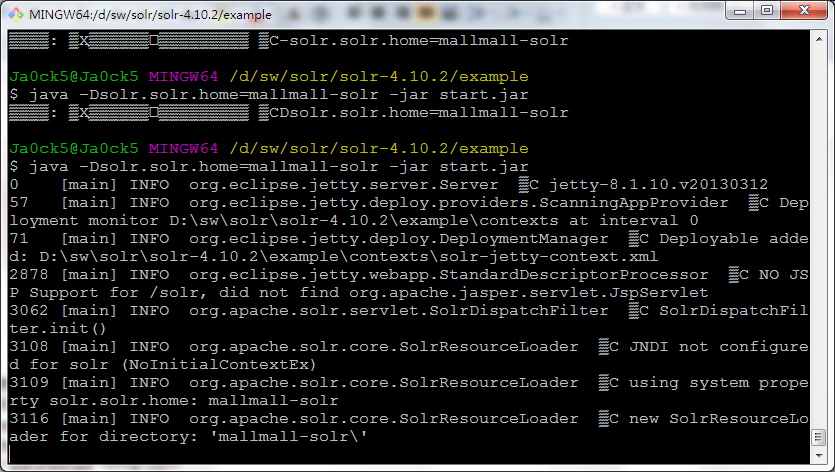
<str name="queryFieldType">string</str>

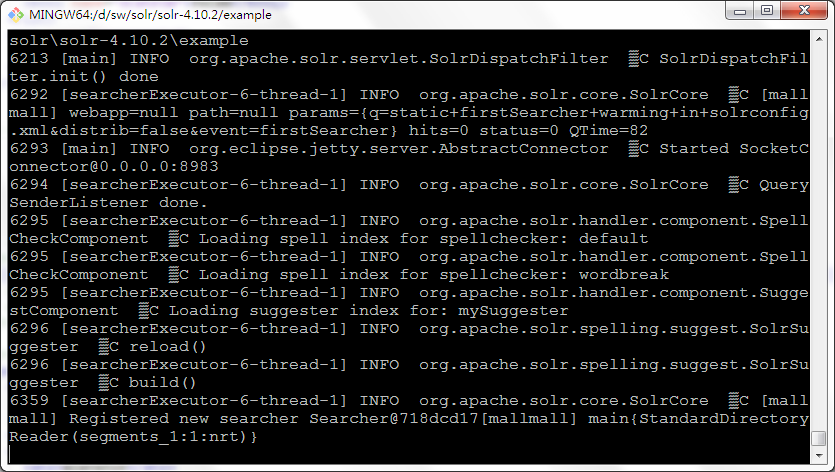
<str name="config-file">elevate.xml</str>

</searchComponent>

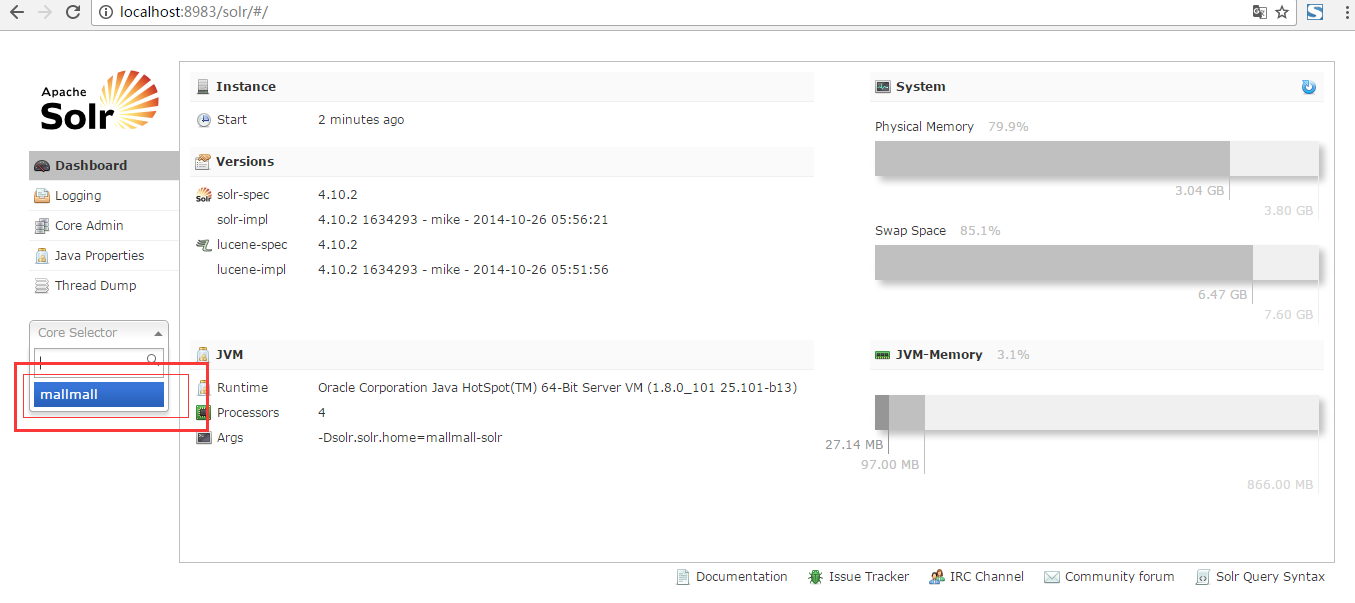
1. 启动自定义的 solr 项目

java -Dsolr.solr.home=mallmall-solr -jar start.jar

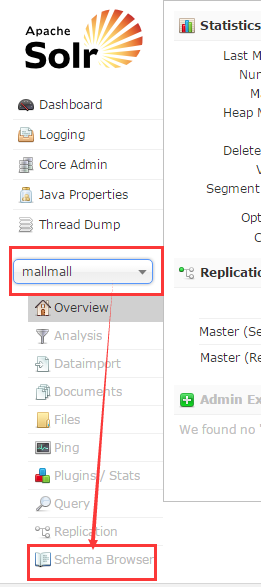


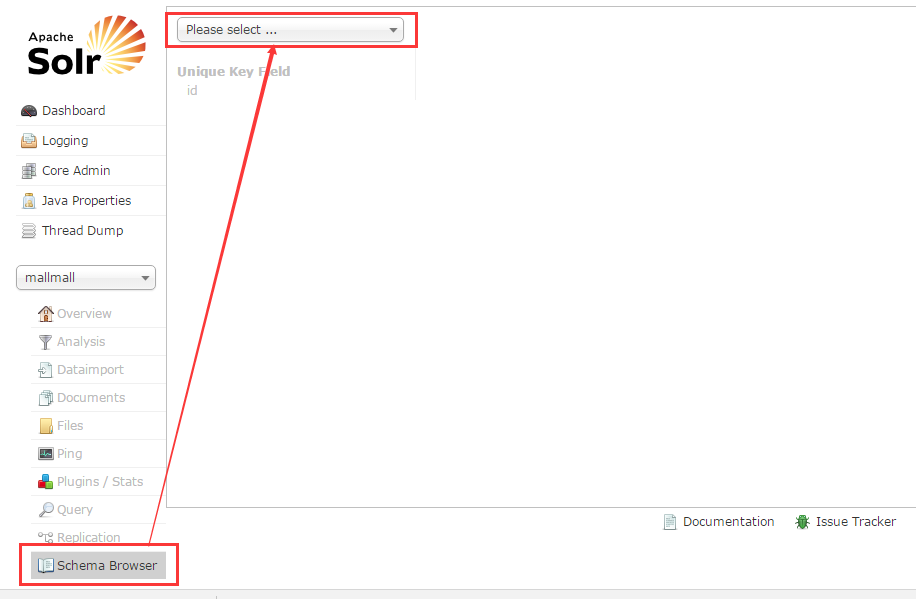


测试启动项目：

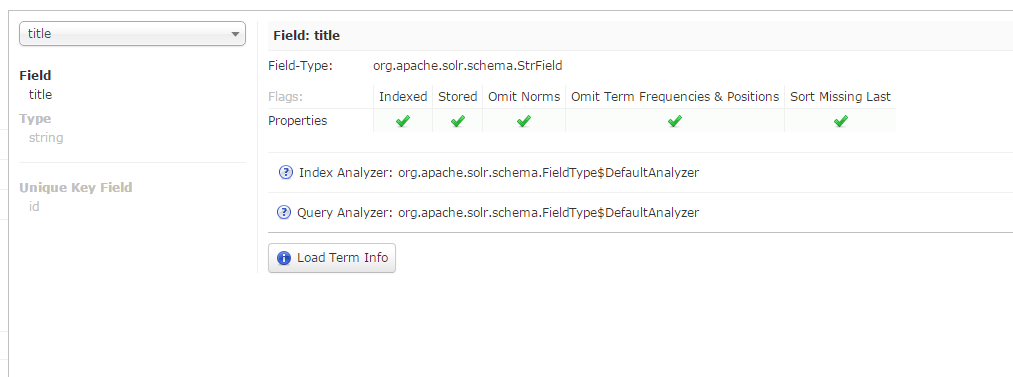


选择：Schema Browser

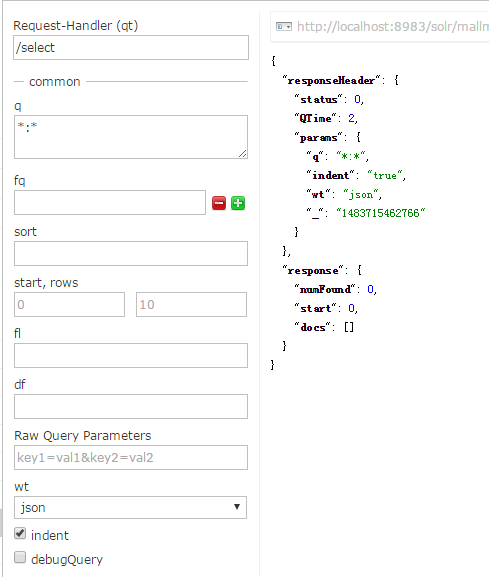




查看刚刚定义的默认查询字段：title



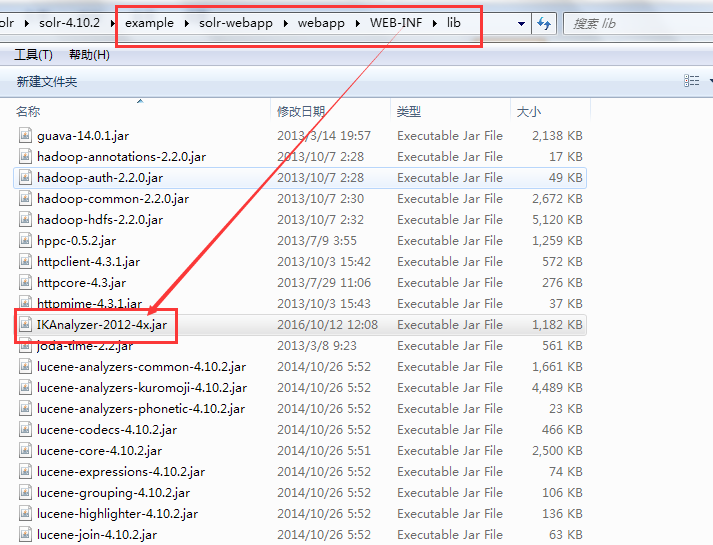
查询：



至此，项目创建成功。

## 集成 IK 分词

将 jar 包放到example\solr-webapp\webapp\WEB-INF\lib



修改 schema.xml

<?xml version="1.0" encoding="UTF-8" ?>

<schema name="example" version="1.5">

<field name="\_version\_" type="long" indexed="true" stored="true"/>

<field name="\_root\_" type="string" indexed="true" stored="false"/>

<!--根据自己表结构修改字段 -->

<field name="id" type="long" indexed="true" stored="true" required="true" multiValued="false" />

<!-- 对应分词器-->

<field name="title" type="text\_ik" indexed="true" stored="true" />

<field name="sellPoint" type="string" indexed="false" stored="true" />

<field name="price" type="long" indexed="true" stored="true" />

<field name="image" type="string" indexed="false" stored="true" />

<field name="cid" type="long" indexed="true" stored="true" />

<field name="status" type="int" indexed="true" stored="false" />

<uniqueKey>id</uniqueKey>

<fieldType name="string" class="solr.StrField" sortMissingLast="true" />

<fieldType name="int" class="solr.TrieIntField" precisionStep="0" positionIncrementGap="0"/>

<fieldType name="long" class="solr.TrieLongField" precisionStep="0" positionIncrementGap="0"/>

<!-- 分词器-->

<fieldType name="text\_ik" class="solr.TextField">

<analyzer class="org.wltea.analyzer.lucene.IKAnalyzer"/>

</fieldType>

</schema>

测试中文分词;

