

2017-04-06 solr使用指南

第一部分：了解solr

一、solr是什么？

Solr 是一个开源的企业级搜索服务器，底层使用易于扩展和修改的Java 来实现。服务器通信使用标准的HTTP 和XML，所以如果使用Solr 了解Java 技术会有用却不是必须的要求。

二、lucene是什么？

Lucene 是一个基于 Java 的全文信息检索工具包，它不是一个完整的搜索应用程序，而是为你的应用程序提供索引和搜索功能。Lucene 目前是 Apache Jakarta 家族中的一个开源项目。也是目前最为流行的基于 Java 开源全文检索工具包。

目前已经有很多应用程序的搜索功能是基于 Lucene，比如 Eclipse 帮助系统的搜索功能。Lucene 能够为文本类型的数据建立索引，所以你只要把你索引的数据格式转化的文本格式，Lucene 就能对你的文档进行索引和搜索。

三、Solr VS Lucene

Solr 与Lucene 并不是竞争对立关系，恰恰相反Solr 依存于Lucene，因为Solr 底层的核心技术是使用Apache Lucene 来实现的，简单的说Solr 是Lucene 的服务器化。需要注意的是Solr 并不是简单的对Lucene 进行封装，它所提供的大部分功能都区别于Lucene。

第二部分 入门教程

一、从网站上下载

官方网站：<http://lucene.apache.org/solr/>

我找的版本是5.3.1，最新版本是5.5

二、安装与运行

1、环境要求

java的版本大于 1.7（利用java -version查看）

php版本 >= 5.2.11

2、启动：

```
bin/solr start -e cloud -noprompt
```

含义：

-e: <example>

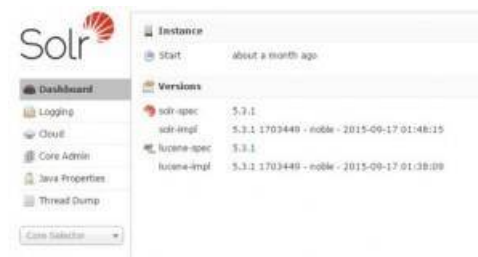
cloud: solrcloud example

-noprompt 对输入不进行提示，接受所有默认输入

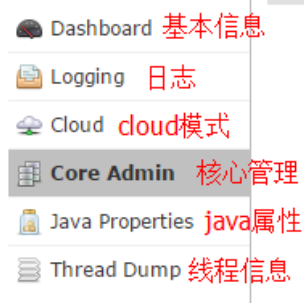
默认是8983端口

查看地址：<http://localhost:8983/solr/>

2.1 效果如图:



2.2 命令含义:



2.3 其他命令:

`bin/solr start -p 8984` (指定为8984端口)

`bin/solr create -c` 指定一个实例

`bin/solr create -help` 帮助

3、建立一个简单实例:

`bin/post -c gettingstarted docs/`

gettingstarted: 索引的名字

docs: 数据

4、怎么进行搜索?

4.1、管理后台搜索

`http://localhost:8983/solr/#/gettingstarted_shard1_replica1/query`

搜索结果如图

The screenshot shows the Solr Admin interface. On the left is a sidebar with navigation links: Dashboard, Logging, Core Admin, Java Properties, Thread Dump, a dropdown menu with 'test' selected, Overview, Analysis, Dataimport, Documents, Files, Ping, Plugins / Stats, Query (highlighted), Replication, Schema Browser, and Segments info. The main area is titled 'Request-Handler (qt)' and shows a search for '/select'. The query '幸福来得有点不知所措' is entered in the 'q' field. The 'fq' field is empty. The 'sort' field is empty. The 'start, rows' fields are set to '0' and '10'. The 'fl' field is empty. The 'df' field is empty. The 'Raw Query Parameters' section shows 'key1=val1&key2=val2'. The 'wt' dropdown is set to 'json'. The 'indent' checkbox is checked. The 'debugQuery' checkbox is unchecked. On the right, the JSON response is displayed for the URL 'http://192.168.0.254:8983/solr/test/select'. The response includes a status of 0, a QTime of 613, and two documents. The first document has an id of '10000478' and a title of '幸福来得有点不知所措'. The second document has an id of '2907412' and a title of '因为有你所以觉得幸福'.

4.2、直接请求:

[http://localhost:8983/solr/gettingstarted/select?](http://localhost:8983/solr/gettingstarted/select?wt=json&indent=true&q=foundation)

[wt=json&indent=true&q=foundation](http://localhost:8983/solr/gettingstarted/select?wt=json&indent=true&q=foundation)

4.3、利用api

第三部分 进阶教程

快速入门主要讲的是solr管理界面，并且已经利用给好的例子做简单的搜索。接下来做的是利用数据库是数据来建议搜索。

一、目录说明

bin 常用命令脚本

contrib 各种jar包

dist 各种jar包

server web服务器

solr 未来创建的core会在该目录下

configsets solr配置集,新建的core可以从这里拷贝配置

二、创建一个搜索实例

bin/solr create -c test

三、配置分词功能

目前sphinx用的是mmseg分词，而solr支持的分词支持较多，这里选择smartcn(solr自带)。

1、导入smartcn的jar包

在{solr安装路径}/server/solr/test/conf/solrconfig.xml加入如下代码

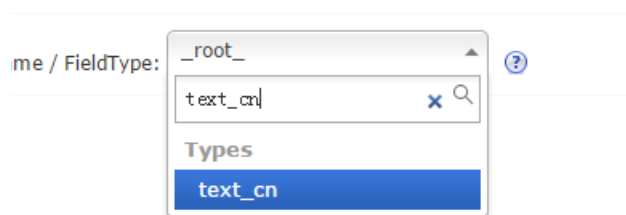
```
<lib dir="${solr.install.dir:../../..}/contrib/analysis-extras/lucene-libs/"
  regex=".*smartcn.*\.jar"></lib>
```

2、配置分词器

在[solr安装路径]/server/solr/test/conf/schema.xml加入如下代码

```
<fieldType name="text_cn" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer
class="org.apache.lucene.analysis.cn.smart.SmartChineseSentenceTokenizerFactory"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter
class="org.apache.lucene.analysis.cn.smart.SmartChineseWordTokenFilterFactory"/>
  </analyzer>
</fieldType>
```

3、效果



分词效果：这里的好处就是可以直接界面测试。之前遇到的一个梗，就是123456qq分词的结果一般都是123456和qq，所以搜123456q是搜不到的

测试效果如下：

WTF	text	123456	qq
	raw_bytes	[31 32 33 34 35 36]	[71 71]
	start	0	6
	end	6	8
	positionLength	1	1
	type	word	word
	position	1	2

四、配置导入功能

1、导入相关jar包

①mysql的jar包

下载地址：<https://dev.mysql.com/downloads/connector/j/>

将jar放置到[solr安装路径]/dist目录下

在[solr安装路径]/server/solr/test/conf/solrconfig.xml加入如下代码

```
<lib dir="${solr.install.dir}../../../../dist/" regex="mysql.*\.jar" />
```

②dataimporthandler包

在[solr安装路径]/server/solr/test/conf/solrconfig.xml加入如下代码

```
<lib dir="${solr.install.dir}../../../../dist/" regex="solr-dataimporthandler.*\.jar" />
```

2、配置handler

在[solr安装路径]/server/solr/test/conf/solrconfig.xml加入如下代码

```
<requestHandler name="/dataimport" class="solr.DataImportHandler">
  <lst name="defaults">
```

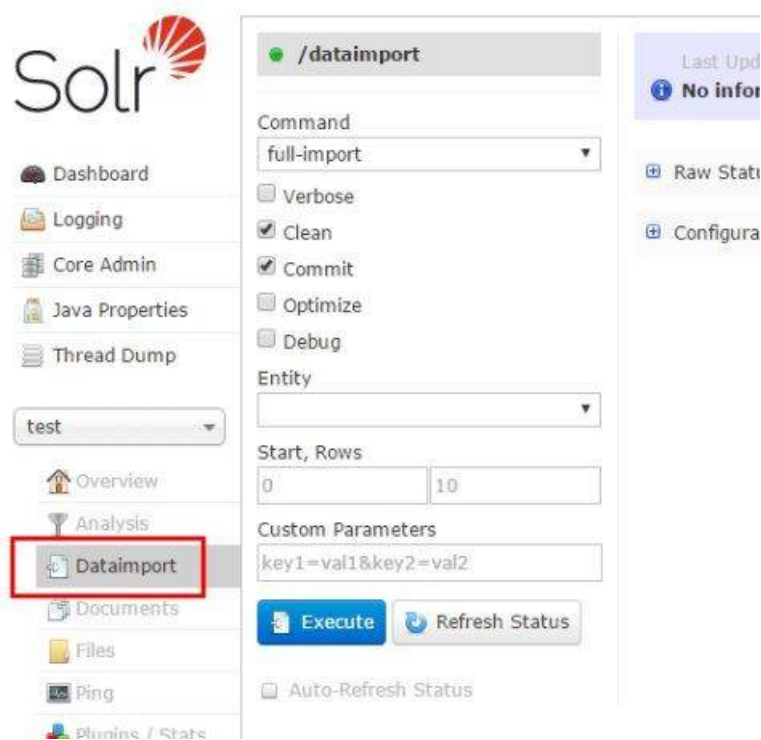
```
<str name="config">db-data-config.xml</str>
</lst>
</requestHandler>
```

3、配置数据源

在[solr安装路径]/server/solr/test/conf/下新建db-data-config.xml,配置如下:

```
<dataConfig>
  <dataSource driver="com.mysql.jdbc.Driver" url="jdbc:mysql://127.0.0.1:3306/test" user="root"
password="root"/>
  <document name="articles">
    <entity name="cms_article" query="select id,title,content,create_time from cms_article">
      <field column="id" name="id" />
      <field column="title" name="title" />
      <field column="content" name="content" />
      <field column="create_time" name="create_time" />
    </entity>
  </document>
</dataConfig>
```

4、导入数据



五、效率问题

10w的数据



100w的数据



第四部分 常见QA

1、为什么不用elasticsearch?

搭建需要1G的内存，成本很高。

PS:可以尝试低版本。

2、sphinx和solr的效率对比

根据sphinx的记录，sphinx效率是5-8w docs/sec

solr是基于java单纯执行速度上比C写的sphinx慢

3、重启后core消失了。

使用6.x版本

<https://www.codeenigma.com/host/faq/how-do-i-create-solr-core-my-server#solr4>

4、节点（node）= solr实例 + 数据

solr实例 可以 包含多个核心

5、