



Бесплатная электронная книга

УЧУСЬ

solr

Free unaffiliated eBook created from
Stack Overflow contributors.

#solr

.....	1
1: solr	2
.....	2
Examples.....	2
.....	2
SOLR.....	4
.....	4
.....	4
SOLR.....	6
.....	7
.....	8
Solr.....	14
2: Apache Solr	16
Examples.....	16
Solr.....	16
Solr ().....	16
3: fieldType	18
.....	18
Examples.....	18
Solr Java.....	18
.....	19
4: Lucene	21
Examples.....	21
.....	21
.....	21
.....	21
.....	21
.....	21
.....	22
.....	22
.....	22

5: HIVE2 SOLR DIH	24
.....	24
.....	24
Examples.....	24
.....	24
6: Solr	26
.....	26
Examples.....	26
?.....	26
.....	28

Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [solr](#)

It is an unofficial and free solr ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official solr.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

глава 1: Начало работы с solr

замечания

В этом разделе представлен обзор того, что такое solr, и почему разработчик может захотеть его использовать.

Он также должен упомянуть о любых крупных предметах в solr и ссылаться на связанные темы. Поскольку документация для solr нова, вам может потребоваться создать начальные версии этих связанных тем.

Examples

Вступление

Solr - автономный поисковый сервер предприятия с REST-подобным API. Вы помещаете в него документы (называемые «индексацией») через JSON, XML, CSV или двоичные файлы через HTTP. Вы запрашиваете его через HTTP GET и получаете JSON, XML, CSV или бинарные результаты. Solr использует библиотеку поиска Lucene и расширяет ее.

Вот некоторые из основных функций, которые предоставляет solr:

- Расширенные возможности полнотекстового поиска

Работает от Lucene™, Solr обеспечивает мощные возможности согласования, включая фразы, подстановочные знаки, объединения, группировку и многое другое по любому типу данных

- **Оптимизирован для трафика большого объема**

Solr доказана в чрезвычайно больших масштабах во всем мире

- **Открытые интерфейсы на основе стандартов - XML, JSON и HTTP**

Solr использует инструменты, которые вы используете для создания приложений

- **Комплексные интерфейсы администрирования**

Solr поставляется со встроенным отзывчивым административным пользовательским интерфейсом, который упрощает управление вашими экземплярами Solr

- **Легкий мониторинг**

Вам нужно больше узнать о своих примерах? Solr публикует множество метрических данных через JMX

- **Высокая масштабируемость и отказоустойчивость**

Solr позволяет легко масштабироваться вверх и вниз. Solr выпекает репликацию, распределение, перебалансировку и отказоустойчивость из коробки.

- **Гибкий и адаптируемый с простой конфигурацией**

Solr предназначен для адаптации к вашим потребностям, при этом упрощая конфигурацию

- **Рядом с индексом реального времени**

Хотите увидеть свои обновления сейчас? Solr использует возможности Lucene Near Real-Time Indexing, чтобы убедиться, что вы видите свой контент, когда хотите его увидеть.

- **Расширяемая архитектура плагинов**

Solr публикует много четко определенных точек расширения, которые упрощают плагин как плагинов индекса, так и времени запроса. Конечно, поскольку это открытый с Apache открытый источник, вы можете изменить любой код, который вы хотите!

Некоторые интересные функции:

- **Схема, когда вы хотите, схематично, когда вы не**

При запуске запускается схематизированный режим Solr, а затем заблокируйте его, когда придет время для производства.

- **Мощные расширения**

Solr поставляется с дополнительными плагинами для индексирования богатого контента (например, PDF-файлы, Word), определения языка, группировки результатов поиска и т. Д.

- **Граничный поиск и фильтрация**

Вырезайте и кубите свои данные, как вы сочтете нужным, используя большой массив алгоритмов огранки

- **Геопространственный поиск**

Включение поиска на основе местоположения выполняется с помощью встроенной поддержки Solr для пространственного поиска

- **Предложения предложений, правописание и многое другое**

Solr поставляется с передовыми возможностями для автозаполнения (поиск по типам), проверка орфографии и многое другое

- **Богатый анализ документов**

Solr поставляется с встроенным Apache Tika, что позволяет легко индексировать богатый контент, такой как Adobe PDF, Microsoft Word и многое другое.

Установка SOLR

Вступление

Следующая процедура была протестирована на тестовом экземпляре в AWS с Redhat и solr 6.1.0 . Возможно, вам придется соответствующим образом настроить процесс на свою операционную систему и среду.

Предпосылки

1. Убедитесь, что вы используете RedHat или аналогичную (Fedora-based) ОС.

```
cat /etc/redhat-release
```

отображает вашу версию ОС.

```
[ec2-user@ip-172-31-28-149 ~]$ cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.2 (Maipo)
```

2. Проверьте, установлен ли java 1.6 или выше

```
which java
```

```
[ec2-user@ip-172-31-28-149 ~]$ which Java
/usr/bin/which: no Java in (/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/ec2-user/.local/bin:/home/ec2-user/bin)
```

3. При необходимости установите Java

```
sudo yum list available java*
```

```
[ec2-user@ip-172-31-28-149 ~]$ sudo yum list available java*
Loaded plugins: amazon-id, rhui-lb, search-disabled-repos
Available Packages
java-1.6.0-openjdk.x86_64                1:1.6.0.40-1.13.12.5.el7_2
java-1.6.0-openjdk-devel.x86_64        1:1.6.0.40-1.13.12.5.el7_2
java-1.7.0-openjdk.x86_64                1:1.7.0.111-2.6.7.2.el7_2
java-1.7.0-openjdk-devel.x86_64        1:1.7.0.111-2.6.7.2.el7_2
java-1.7.0-openjdk-headless.x86_64     1:1.7.0.111-2.6.7.2.el7_2
java-1.8.0-openjdk.x86_64                1:1.8.0.111-1.b15.el7_2
```

```
sudo yum install java-1.8.0-openjdk.x86_64
```

4. Проверьте правильность установки

`which java` отображает Java-дом

`java -version` отображает `java -version Java`

```
[ec2-user@ip-172-31-28-149 ~]$ which java
/usr/bin/java
[ec2-user@ip-172-31-28-149 ~]$ java -version
openjdk version "1.8.0_111"
OpenJDK Runtime Environment (build 1.8.0_111-b15)
OpenJDK 64-Bit Server VM (build 25.111-b15, mixed mode)
```

5. Создайте пользователя SOLR.

```
sudo adduser solr
```

6. Добавьте пароль для пользователя.

```
sudo passwd solr
```

```
[ec2-user@ip-172-31-28-149 ~]$ sudo adduser solr
[ec2-user@ip-172-31-28-149 ~]$ sudo passwd solr
Changing password for user solr.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

7. Включить sudo для пользователя SOLR, запустить visudo

```
sudo visudo
```

8. Найдите следующие строки:

```
## Allows people in group wheel to run all commands
# %wheel ALL=(ALL) ALL
```

9. Если %wheel закомментировано, раскомментируйте вторую строку, удалив символ #.

```
%wheel ALL=(ALL) ALL
```

```
## Allows people in group wheel to run all commands
%wheel ALL=(ALL) ALL
```

10. Если вы внесли изменения, используйте :wq иначе используйте :q для выхода.

11. Добавьте пользователя solr в группу колес.

```
sudo usermod -aG wheel solr
```

12. Перейдите к пользователю solr проверьте, есть ли у вас права root:

```
su solr -
sudo whoami
```



```
[ec2-user@ip-172-31-28-149 ~]$ sudo usermod -aG wheel solr
[ec2-user@ip-172-31-28-149 ~]$ su solr
Password:
[solr@ip-172-31-28-149 ec2-user]$ groups
solr wheel
[solr@ip-172-31-28-149 ec2-user]$ sudo whoami

We trust you have received the usual lecture from the local
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for solr:
root
```

Загрузка SOLR

13. Найдите свое местное зеркало по адресу:

```
http://www.apache.org/dyn/closer.lua/lucene/solr/
```

14. Измените местоположение своего каталога, чтобы выбрать:

```
cd /opt/
```

15. Загрузите копию пакета из зеркала:

```
sudo curl -O http://www.trieuvan.com/apache/lucene/solr/6.1.0/solr-6.1.0.tgz
```

```
[solr@ip-172-31-28-149 opt]$ sudo curl -O http://www.trieuvan.com/apache/lucene/solr/6.2.0/solr-6.2.0.tgz
```

% Total Time	% Received Current	% Xferd	Average Speed	Time	Time
Left	Speed	Dload	Upload	Total	Spent
0	0	0	0	0	0
3	142M	3	5343k	0	0
9	142M	9	13.5M	0	0
13	142M	13	19.3M	0	0
16	142M	16	23.4M	0	0
19	142M	19	27.4M	0	0

16. Упаковать пакет:

```
sudo tar zxvf solr-6.1.0.tgz
```

17. Скопируйте сценарий установщика в папку:

```
sudo cp /opt/solr-6.1.0/bin/install_solr_service.sh .
```

18. Удалите ненужные файлы:

```
sudo rm -rf solr-6.1.0
```

устанавливать

1. Запустите скрипт установки:

```
sudo ./install_solr_service.sh solr-6.1.0.tgz
```

```
Started Solr server on port 8983 (pid=9607). Happy searching!

Found 1 Solr nodes:

Solr process 9607 running on port 8983
{
  "solr_home":"/var/solr/data",
  "version":"6.2.0 764d0f19151dbff6f5fcd9fc4b2682cf934590c5 - mi
ke - 2016-08-20 05:41:37",
  "startTime":"2016-10-20T04:02:37.028Z",
  "uptime":"0 days, 0 hours, 0 minutes, 15 seconds",
  "memory":"85.4 MB (%17.4) of 490.7 MB"}

Service solr installed.
```

2. Сделать автозапуск службы SOLR при перезагрузке сервера.

```
sudo chkconfig --add solr
```

```
chkconfig | grep solr
```

```
[solr@ip-172-31-28-149 ec2-user]$ sudo chkconfig --add solr
[sudo] password for solr:
[solr@ip-172-31-28-149 ec2-user]$ chkconfig |grep solr

Note: This output shows SysV services only and does not include native
systemd services. SysV configuration data might be overridden by native
systemd configuration.

If you want to list systemd services use 'systemctl list-unit-files'.
To see services enabled on particular target use
'systemctl list-dependencies [target]'.

solr          0:off  1:off  2:on   3:on   4:on   5:off
n             6:off
```

3. Изменить владельца услуги

```
sudo chown -R solr:solr /var/solr/
```

Тестирование вашей установки

1. Создайте ядро из командной строки:

```
sudo su - solr -c "/opt/solr/bin/solr create -c NewCore1 -n data_driven_schema_configs"
```

```
[solr@ip-172-31-28-149 ec2-user]$ sudo su - solr -c "/opt/solr/bin/solr create -c NewCore1 -n data_driven_schema_configs"
```

```
Copying configuration to new core instance directory:  
/var/solr/data/NewCore1
```

```
Creating new core 'NewCore1' using command:
```

```
http://localhost:8983/solr/admin/cores?action=CREATE&name=NewCore1&instanceDir=NewCore1
```

```
{  
  "responseHeader": {  
    "status": 0,  
    "QTime": 2856},  
  "core": "NewCore1"}  
}
```

2. Откройте Администратор в браузере:

```
http://<solr_server>:8983/solr/#/
```

3. Откройте список ядер в меню, чтобы увидеть ядро NewCore1.

Statistics

- Last Modified: -
- Num Docs: 0
- Max Doc: 0
- Heap Memory 0
- Usage: Deleted Docs: 0
- Version: 2
- Segment Count: 0
- Optimized: ✓
- Current: ✓

Replication (Master)

	Version	Gen	Size
Master (Searching)	0	1	71 bytes
Master (Replicable)	-	-	-

+ Admin Extra

4. Проверьте, являются ли сердечники липкими:

```
sudo service solr restart
```

5. Обновите администратор в браузере:

http://<solr_server>:8983/solr/#/ Убедитесь, что страница администратора перезагружается, а ядро появляется после перезагрузки.

6. Просмотр состояния сервера в командной строке:

```
sudo service solr status
```

7. Подготовьте новую конфигурацию ядра для создания ядра в веб-админере, изменив местоположение вашего каталога на данные:

```
cd var/solr/data/
```

```

[solr@ip-172-31-28-149 bin]$ pwd
/opt/solr/bin
[solr@ip-172-31-28-149 bin]$ cd ../../..
[solr@ip-172-31-28-149 /]$ pwd
/
[solr@ip-172-31-28-149 /]$ cd var/solr/data/
[solr@ip-172-31-28-149 data]$ ll
total 4
drwxrwxr-x. 4 solr solr  50 Oct 20 22:54 NewCore1
-rw-r-----. 1 solr solr 2117 Oct 20 00:02 solr.xml
[solr@ip-172-31-28-149 data]$ ll NewCore1/
total 8
drwxrwxr-x. 3 solr solr 4096 Oct 20 00:02 conf
-rw-rw-r--. 1 solr solr  78 Oct 20 22:54 core.properties
drwxrwxr-x. 5 solr solr  53 Oct 20 22:54 data

```

8. Здесь хранятся новые ядра:

```
ll
```

9. В качестве шаблона может быть использована папка conf вновь созданного ядра.

```
ll NewCore1/
```

10. Создайте папку для другого ядра, которое вы создадите в веб-админе:

```
mkdir CoreFromWebAdmin
```

11. Скопируйте каталог conf в новое место:

```
sudo cp -R NewCore1/conf/ CoreFromWebAdmin
```

12. Переключиться на веб-интерфейс администратора в браузере

13. Нажмите «Добавить ядро»

14. Добавьте CoreFromWebAdmin в качестве имени и папки для нового ядра.

Solr Admin interface showing the 'Add Core' dialog box. The dialog contains the following fields:

- name: CoreFromWebAdmin
- instanceDir: CoreFromWebAdmin
- dataDir: data
- config: solrconfig.xml
- schema: schema.xml

Message: instanceDir and dataDir need to exist before you can create the core

Buttons: Add Core (checked), Cancel

Terminal output (background):

```
solr@ip-172-31-10-100 ~
[ solr@ip-172-31-10-100 ~]# ls -ld /var/lib/solr/core0/eFromWebAdmin/
total 4
drwxrwxr-x. 4 solr solr 4096 Aug 11 12:54
-rw-r-----. 1 solr solr 4096 Aug 11 12:54
[ solr@ip-172-31-10-100 ~]# ls -ld /var/lib/solr/core0/eFromWebAdmin/
total 8
drwxrwxr-x. 3 solr solr 4096 Aug 11 12:54
-rw-rw-r--. 1 solr solr 4096 Aug 11 12:54
drwxrwxr-x. 5 solr solr 4096 Aug 11 12:54
[ solr@ip-172-31-10-100 ~]# ls -ld /var/lib/solr/core0/eFromWebAdmin/
total 4
drwxrwxr-x. 3 solr solr 4096 Aug 11 12:54
[ solr@ip-172-31-10-100 ~]# ls -ld /var/lib/solr/core0/eFromWebAdmin/
total 4
drwxr-xr-x. 3 solr solr 4096 Aug 11 12:54
[ solr@ip-172-31-10-100 ~]# ls -ld /var/lib/solr/core0/eFromWebAdmin/
total 4
drwxr-xr-x. 3 solr solr 4096 Aug 11 12:54
[ solr@ip-172-31-10-100 ~]#
```

15. Откройте новое ядро.

16. Щелкните документы, чтобы добавить документы.

17. Выберите формат XML и вставьте следующий код:

```
<add><doc>
  <field name="id">F9V7464-APL-KIT</field>
  <field name="name">Belkin Mobile Power Cord for iPod w/ Dock</field>
  <field name="manu">Belkin</field>
  <!-- Join -->
  <field name="manu_id_s">belkin</field>
  <field name="cat">electronics</field>
  <field name="cat">connector</field>
  <field name="features">car power adapter, white</field>
  <field name="weight">4.0</field>
  <field name="price">19.95</field>
  <field name="popularity">1</field>
  <field name="inStock">>false</field>
  <!-- Buffalo store -->
  <field name="store">45.18014,-93.87741</field>
</doc></add>
```

```

<field name="manufacturedate_dt">2005-08-01T16:30:25Z</field>
</doc>
<doc>
  <field name="id">IW-032</field>
  <field name="name">iPod & amp; iPod Mini USB 2.0 Cable</field>
  <field name="manu">Belkin</field>
  <!-- Join -->
  <field name="manu_id_s">belkin</field>
  <field name="cat">electronics</field>
  <field name="cat">connector</field>
  <field name="features">car power adapter for iPod, white</field>
  <field name="weight">2.0</field>
  <field name="price">11.50</field>
  <field name="popularity">1</field>
  <field name="inStock">>false</field>
  <!-- San Francisco store -->
  <field name="store">37.7752,-122.4232</field>
  <field name="manufacturedate_dt">2006-02-14T23:55:59Z</field>
</doc>
<doc>
  <field name="id">F887464-APL-KIT</field>
  <field name="name">Belkin Mobile Power Cord for iPod w/ Dock</field>
  <field name="manu">Belkin</field>
  <!-- Join -->
  <field name="manu_id_s">belkin</field>
  <field name="cat">electronics</field>
  <field name="cat">connector</field>
  <field name="features">car power adapter, black</field>
  <field name="weight">4.0</field>
  <field name="price">19.95</field>
  <field name="popularity">1</field>
  <field name="inStock">>true</field>
  <!-- Buffalo store -->
  <field name="store">45.18014,-93.87741</field>
</doc>
<doc>
  <field name="id">FAV7464-APL-KIT</field>
  <field name="name">Belkin Mobile Power Cord for iPod w/ Dock</field>
  <field name="manu">Belkin</field>
  <!-- Join -->
  <field name="manu_id_s">belkin</field>
  <field name="cat">electronics</field>
  <field name="cat">connector</field>
  <field name="features">car power adapter, blue</field>
  <field name="weight">4.0</field>
  <field name="price">15.95</field>
  <field name="popularity">2</field>
  <field name="inStock">>true</field>
  <!-- Buffalo store -->
  <field name="store">45.18014,-93.87741</field>
  <field name="manufacturedate_dt">2015-09-21T16:30:25Z</field>
</doc></add>

```

Если ваш ответ успешно завершен, вы успешно установили SOLR и проверили вашу установку.



- Dashboard
- Logging
- Core Admin
- Java Properties
- Thread Dump

NewCore1

- Overview
- Analysis
- Dataimport
- Documents
- Files
- Ping
- Plugins / Stats
- Query
- Replication
- Schema
- Segments info

Request-Handler (qt)

/update

Document Type

XML

Document(s)

```
<add>
<doc>
  <field name="id">F9V7464-APL-KIT</field>
  <field name="name">Belkin Mobile Power Cord for iPod w/ Dock</field>
  <field name="manu">Belkin</field>
  <!-- Join -->
  <field name="manu_id_s">belkin</field>
  <field name="cat">electronics</field>
  <field name="cat">connector</field>
  <field name="features">car power adapter, white</field>
</doc>
</add>
```

Commit Within

1000

Overwrite

true

Submit Document

Status: success
Response:
{
 "response": {
 "status": "OK"
 }
}

Пример поиска Solr



- Dashboard
- Logging
- Core Admin
- Java Properties
- Thread Dump
- demo
- Overview
- Analysis
- Dataimport
- Documents
- Files
- Ping
- Plugins / Stats
- Query
- Replication
- Schema
- Segments info

Request-Handler (qt)

/select

— common —

q

:

fq

publisher_s="Bantam"

sort

pubyear_i desc

start, rows

0 3

fl

title_t, pubyear_i

df

Raw Query Parameters

key1=val1&key2=val2

wt

json

indent

debugQuery

dismax

```
http://localhost:8983/solr/demo/select?fl=title_t, puby
{
  "responseHeader":{
    "status":0,
    "QTime":1,
    "params":{
      "q":":*",
      "indent":"on",
      "fl":"title_t, pubyear_i",
      "fq":"publisher_s=\\"Bantam\\"",
      "sort":"pubyear_i desc",
      "rows":"3",
      "wt":"json",
      "_":"1498631813496"}},
  "response":{"numFound":5,"start":0,"docs":[
    {
      "pubyear_i":1999,
      "title_t":["A Clash of Kings"]},
    {
      "pubyear_i":1996,
      "title_t":["A Game of Thrones"]},
    {
      "pubyear_i":1992,
      "title_t":["Snow Crash"]}]}
}}
```

Прочитайте Начало работы с solr онлайн: <https://riptutorial.com/ru/solr/topic/1015/начало-работы-с-solr>

глава 2: Apache Solr

Examples

Установка Solr

Вы можете установить Solr в любой системе, где доступна подходящая Java Runtime Environment (JRE), как описано ниже. В настоящее время это включает Linux, OS X и Microsoft Windows. Вам потребуется Java Runtime Environment (JRE) версии 1.8 или выше. В командной строке проверьте свою версию Java следующим образом:

```
$ java -version
```

Solr можно получить на веб-сайте Solr по адресу <http://lucene.apache.org/solr/> . Извлеките архив распространения Solr в локальный каталог

Быстрый запуск Solr (запуск и остановка солнечной батареи)

Вы можете запустить Solr, выполнив `bin / solr` из каталога Solr, и это запустит Solr в фоновом режиме, прослушивая порт 8983.

```
$ bin / solr start
```

Чтобы изменить порт Solr, вы можете использовать параметр `-p` при запуске, например:

```
$ bin / solr start -p 8984
```

Поскольку Solr является сервером, чаще всего он запускается в фоновом режиме, особенно в Unix / Linux. Однако, чтобы запустить Solr на переднем плане, просто выполните:

```
$ bin / solr start -f
```

При запуске Solr на переднем плане (с использованием `-f`) вы можете остановить его с помощью `Ctrl-c`. Однако при работе в фоновом режиме вы должны использовать команду остановки, например:

```
$ bin / solr stop -p 8983
```

Команда `stop` требует, чтобы вы указывали, какой порт Solr прослушивает или вы можете использовать параметр `-all`, чтобы остановить все запущенные экземпляры Solr.

Если вы не уверены, что Solr работает локально, вы можете использовать команду `status`:

```
Состояние $ bin / solr
```

Прочитайте Apache Solr онлайн: <https://riptutorial.com/ru/solr/topic/9635/apache-solr>

глава 3: Как создать пользовательский fieldType

замечания

Замечания о пользовательском поле класса Java:

Это небольшой раздел большой [сортировки](#) статей в [Solr с использованием внешнего поля](#), написанного для сортировки документов Solr на основе пользовательского полевого компаратора.

Замечания о настраиваемом поле, созданное из существующих полей Solr:

Apache создал подробную документацию по этой теме - « [Понимание анализаторов, токенизаторов и фильтров](#) » .

Examples

Создайте собственный тип поля Solr из собственного пользовательского класса Java

Изменения схемы:

Вам нужно будет определить новый тип поля в вашем файле схемы solr, а затем вы можете создавать поля этого типа. Пример фрагмента схемы:

```
<!-- Source: solr/example/.../conf/schema.xml -->
<?xml version="1.0" encoding="UTF-8" ?>
<schema name="adam" version="1.3">
  <types>
    ...
    <fieldType name="rank_t" class="org.apache.solr.schema.ext.RankFieldType"/>
  </types>
</fields>
  ...
  <field name="rank" type="rank_t" indexed="true" stored="true"/>
</fields>
  ...
</schema>
```

Класс Java для настраиваемого типа поля:

```
// Source: src/java/org/apache/solr/schema/ext/RankFieldType.java
package org.apache.solr.schema.ext;

import java.io.IOException;
```

```

import org.apache.lucene.document.Fieldable;
import org.apache.lucene.search.SortField;
import org.apache.solr.response.TextResponseWriter;
import org.apache.solr.schema.FieldType;
import org.apache.solr.schema.SchemaField;
import org.apache.solr.search.ext.RankFieldComparatorSource;

public class RankFieldType extends FieldType {

    @Override
    public SortField getSortField(SchemaField field, boolean top) {
        return new SortField(field.getName(), new RankFieldComparatorSource(), top);
    }

    @Override
    // copied verbatim from GeoHashField method
    public void write(TextResponseWriter writer, String name, Fieldable f) throws IOException
    {
        writer.writeStr(name, f.stringValue(), false);
    }
}

```

Создание настраиваемого типа поля из доступных типов полей

Давайте перейдем к примеру с некоторыми теоретическими знаниями. Здесь используются три важных термина: [анализаторы](#) , [токенизаторы](#) и [фильтры](#) . Чтобы создать такое настраиваемое поле, вам необходимо создать анализатор с одним токенизатором и одним или несколькими фильтрами. Как уже упоминалось [здесь](#) , вы можете иметь только один Tokenizer каждого анализатора , но есть способы , чтобы преодолеть это ограничение.

```

<fieldType name="alphaOnlySort" class="solr.TextField" sortMissingLast="true"
omitNorms="true">
  <analyzer>
    <tokenizer class="solr.KeywordTokenizerFactory"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.TrimFilterFactory"/>
    <filter class="solr.PatternReplaceFilterFactory" replace="all" replacement=""
pattern="([\^a-z])"/>
  </analyzer>
</fieldType>

```

Другой пример:

```

<fieldType name="lowercase_text" class="solr.TextField" positionIncrementGap="150">
  <analyzer>
    <tokenizer class="solr.KeywordTokenizerFactory" />
    <filter class="solr.LowerCaseFilterFactory" />
  </analyzer>
</fieldType>

```

Еще один пример с описанием:

```

<fieldType name="text_stem" class="solr.TextField">

```

```
<analyzer>
  <tokenizer class="solr.StandardTokenizerFactory"/>
  <filter class="solr.StandardFilterFactory"/>
  <filter class="solr.LowerCaseFilterFactory"/>
  <filter class="solr.EnglishPorterFilterFactory"/>
</analyzer>
</fieldType>
```

Этот пример начинается с стандартного токенизатора Solr, который разбивает текст поля на токены. Затем эти токены проходят через стандартный фильтр Solr, который удаляет точки из аббревиатур и выполняет несколько других общих операций. Затем все маркеры устанавливаются в нижнем регистре, что облегчит совпадение с учетом регистра во время запроса. Последний фильтр в приведенном выше примере представляет собой фильтр стволых фильтров, в котором используется алгоритм строгания Портера. Штокмер - это в основном набор правил отображения, который отображает различные формы слова обратно на основание или слово, из которого они выводятся. Например, на английском языке слова «объятия», «обнимания» и «обнимания» - это все формы слова «обнять». Штокмер заменит все эти термины «объятием», который будет проиндексирован. Это означает, что запрос «hug» будет соответствовать термину «обниматься», но не «огромным».

Пример использования такого настраиваемого поля:

```
<field name="keywords" type="text_stem" indexed="true" stored="true" />
```

Список доступных типов токенизаторов: [список типов токенизаторов](#)

Список доступных типов фильтров: [список типов фильтров](#)

Прочитайте [Как создать пользовательский fieldType онлайн:](#)

<https://riptutorial.com/ru/solr/topic/7409/как-создать-пользовательский-fieldtype>

глава 4: Синтаксис запроса Lucene

Examples

Поиск близости

```
name:"john doe"~1
```

Ищет несколько терминов на определенном расстоянии (~ 1), т. **Е. Найдет** текст, содержащий **john anonymous doe**, но не **john второе имя doe**

Основной поиск

```
name:john
```

Ищет один термин (joe) в одном поле (имя)

Логический поиск

```
+firstname:john +surname:doe
```

Совпадает с документами, где первое имя - john, а фамилия - doe. + prefix указывает, что *должен* появиться поисковый запрос (AND).

```
+firstname:john -surname:doe
```

Совпадает с документами, где firstname - john, а фамилия - не doe. - prefix указывает, что поисковый запрос *не должен* выполняться (NOT).

```
+firstname:john surname:(doe bloggs)
```

Совпадает с документами, где firstname является john, а фамилия - либо doe, либо bloggs. Префикс не указывает, что *должна* произойти фамилия (ИЛИ)

Поиск по фразе

```
name:"john doe"
```

Ищет несколько терминов в определенном порядке.

Расширение поисковых запросов

```
name:(john doe^5)
```

Индикатор ^ можно использовать для увеличения поискового термина, чтобы повысить его уровень релевантности, означающий, что документы, содержащие *лань*, более актуальны, чем содержащие *джон*

Поиск по шаблону

```
name:john*
```

Индикатор * позволяет вам выполнять поиск подстановочных знаков, соответствующий 0 или более символам после поискового запроса *john*, будет возвращать документы, содержащие john, johnson, john's, johnny и т. Д.

```
name:do?
```

? индикатор позволяет вам выполнять поиск по шаблону с единственным символом в поисковом выражении, будет возвращать документы, содержащие лань, собаку, точку и т. Д.

Поиск диапазона

```
age:[50 TO 60]
```

Соответствует документам, возраст которых составляет от 50 до 60, включая 50 и 60

```
age:{50 TO 60}
```

Соответствует документам, возраст которых составляет от 50 до 60, за исключением 50 и 60

```
age:[* TO 60]
```

Соответствует документам, возраст которых меньше или равен 60

```
age:[50 TO *]
```

Соответствует документам, возраст которых больше или равен 50

```
age:{50 to 60]
```

Вы можете комбинировать фигурные и квадратные скобки. Соответствует документам, возраст которых превышает 50, но меньше или равен 60

Присоединяйся к ядрам

```
{!join from=personid to=id fromIndex=AddressCore}address:Address1
```

Итак, если у вас есть два ядра, которые выглядят так:

PersonCore - id, name

AddressCore - id, address, personid

Это найдет все документы PersonCore по определенному адресу

Прочитайте Синтаксис запроса Lucene онлайн: <https://riptutorial.com/ru/solr/topic/2242/>

глава 5: УКАЗАНИЕ ТАБЛИЦЫ HIVE2 В СОЛНЕЧНОМ ИСПОЛЬЗОВАНИИ SOLR DIH

Вступление

Эта документация обеспечивает способ подключения к улью с использованием SOLR Data Import Handler и индексация данных в SOLR. Это интересная документация, потому что я не мог найти ее через Интернет.

Обработчик в основном обрабатывает более 80 миллионов записей, что означает сильную инфраструктуру с хорошими процессорами и память, безусловно, необходима.

замечания

Пожалуйста, не стесняйтесь обращаться к нам, и мы постараемся помочь как можно больше.

Examples

меры

Сначала мы получили банку hive2 и запустили ее через java для проверки возможности подключения. Затем мы поняли, что используемые банки:

1. Hadoop-синфазного 2.7.0-mapr-1703.jar
2. улья общего 2.1.1-mapr-1703-r1.jar
3. улья-2.1.1 JDBC-mapr-1703-r1-standalone.jar

Если вы используете SOLR Cloud, эти банки должны быть переданы на виртуальную машину, где установлен SOLR, а затем ссылаются на файл solrconfig.xml следующим образом:

Импортировать часть в файл solrconfig.xml

```
<lib dir = "/ users / path_to_folder_with_jar" regex = ". *. jar" />
```

Тогда это самая важная часть: ваша строка подключения к ульям:

Часть подключения

```
<dataConfig> <dataSource name = "ABC" driver = "org.apache.hive.jdbc.HiveDriver" url = "jdbc:hive2: //....connectionString" user = "username" password = "password" />
```

```
<document name = "collection_name">
```

```
<entity name = "collection_lookup" query = "select unique_key как id from table_name">
```

```
</ entity>
```

```
</ document>
```

```
</ dataConfig>
```

Нажать конфиг через zookeeper

```
server / scripts / cloud-scripts / zkcli.sh -zkhost host1: 2181, host2: 2181 -cmd upconfig -confname  
configName -confdir server / solr / configsets / folder /
```

Перейдите по [адресу http: // host: 8983 / solr / # / collection_name / dataimport // dataimport](http://host:8983/solr/#/collection_name/dataimport//dataimport),
затем проверьте debug и сначала проверьте 10 или 20 записей.

Вы увидите, что данные текут. CHEERS !! Я могу помочь, если вы хотите обсудить дальше,
но я предполагаю, что это должно делать. Он работает для меня.

Прочитайте **УКАЗАНИЕ ТАБЛИЦЫ HIVE2 В СОЛНЕЧНОМ ИСПОЛЬЗОВАНИИ SOLR DIH**
онлайн: <https://riptutorial.com/ru/solr/topic/10697/указание-таблицы-hive2-в-солнечном-использовании-solr-dih>

глава 6: Устранение неполадок Solr

замечания

При поиске с целью команды `copyField` используемый стек анализатора является одним из целевого объекта `copyField`, а не источника. Это особенно важно помнить с настройкой по умолчанию, которая выполняет поиск по общему **текстовому** полю.

Examples

Был ли мой контент фактически проиндексирован?

Часто люди пытаются индексировать некоторый контент, а затем найти его. Если они не могут увидеть ожидаемый результат, они пытаются устранить весь сквозной процесс. Лучше всего посмотреть, действительно ли индексируется контент в ожидаемых полях. Таким образом, проблема разбивается на две: индексирование и поиск.

Самый простой способ проверить, что индексируется на экране схемы интерфейса администратора (1). Просто выберите соответствующее поле (2) и загрузите его информацию о терминах (3). Это все индексированные термины в этой области. Список (4) может быть довольно длинным, но вы можете изменить количество отображаемых элементов и / или индексов в отдельное тестовое поле только для отладки.

- Dashboard
- Logging
- Core Admin
- Java Properties
- Thread Dump
- techproducts
- Overview
- Analysis
- Dataimport
- Documents
- Files
- Ping
- Plugins / Stats
- Query
- Replication
- Schema**
- Segments info

2

Field
cat
Copied to
text **x**
Type
string

Unique Key Field
id

Field: cat

Field-Type: org.apache.solr.schema.StrField
Docs: 20

Flags:	Indexed	Tokenized	Stored
Properties	✓		✓
Schema	✓		✓
Index	✓	✓	✓

3

Autoload

10 / 16 Top-Terms

12	electronics
4	currency
3	memory
2	search connector software hard drive graphics card
1	electronics and s electronics and c

Если вы не видите ничего или нет ожидаемого контента, это обычно означает, что индексирование не удалось. Если вы видите контент здесь, но не в запросе, то это поиск, необходимый для устранения неполадок.

Прочитайте Устранение неполадок Solr онлайн: <https://riptutorial.com/ru/solr/topic/1969/устранение-неполадок-solr>

кредиты

S. No	Главы	Contributors
1	Начало работы с solr	Community , esiprogrammer , Lefty G Balogh , mfatihk , SimplyInk
2	Apache Solr	Pralhad Awasthi
3	Как создать пользовательский fieldType	RamenChef , saurav
4	Синтаксис запроса Lucene	BunkerMentality , Fuu , MatsLindh
5	УКАЗАНИЕ ТАБЛИЦЫ NIVE2 В СОЛНЕЧНОМ ИСПОЛЬЗОВАНИИ SOLR DIH	Khalid Imam
6	Устранение неполадок Solr	Alexandre Rafalovitch