



Apache Solr 4.4 Basic Operation Guide

HƯỚNG DẪN CƠ BẢN VỀ APACHE SOLR 4.4

(Version 1.0)

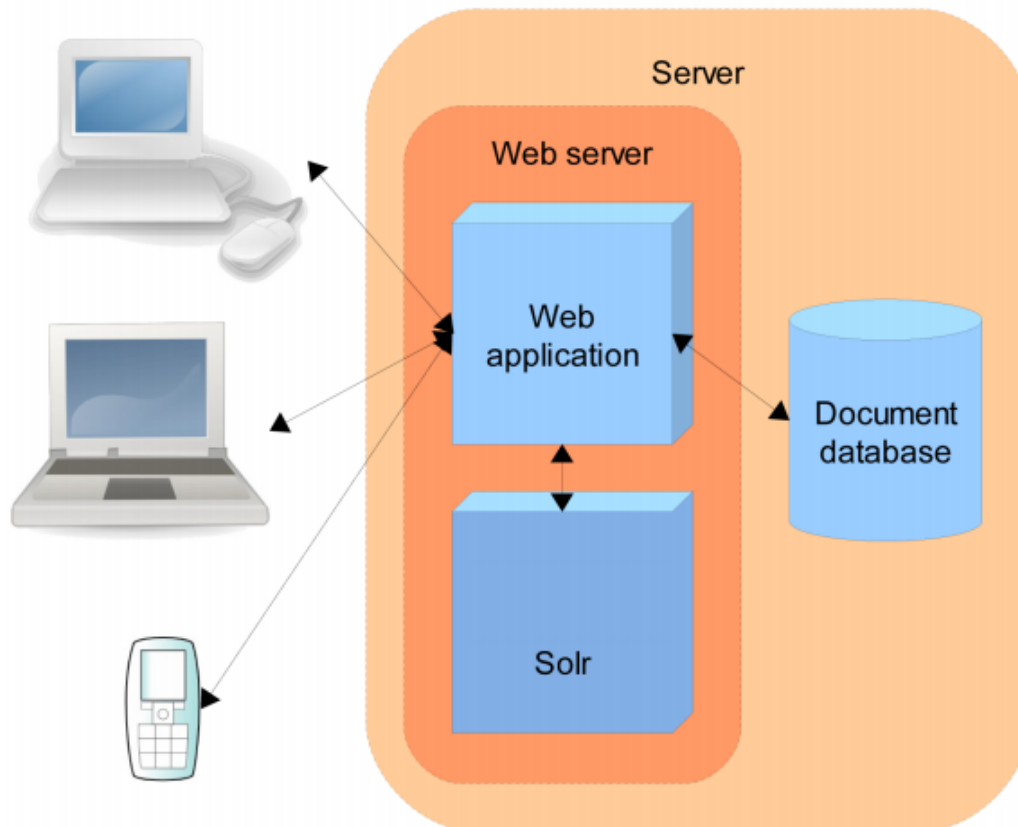
Author: Nguyễn Khánh Linh (Tìm hiểu và tổng hợp)

Table of Contents

I.	Giới thiệu chung	3
II.	Cài đặt.....	3
	Cơ bản	3
	Cài đặt Solr	3
III.	Chạy Solr	4
IV.	Configuration	4
	Cách thêm trường mới	5
	Cách thêm Request Handler	7
V.	Thêm document vào Solr	9
	Chạy bằng Solr admin	10
	Chạy bằng lệnh trên Eclipse/Netbeans	11
VI.	Xoá dữ liệu/document trên Solr Admin.....	12
VII.	Tìm kiếm trên Solr Admin	13

I. Giới thiệu chung

Solr là một nền tảng (platform) tìm kiếm mã nguồn mở. Nó bao gồm các chức năng tìm kiếm như full-text search, hit highlighting, faceted search, near real-time indexing, clustering động, tích hợp database, xử lý rich document (ví dụ Word và Pdf) và geospatial search



In the scenario above, Solr runs alongside another application in a Web server. For example, an online store application would provide a user interface, a shopping cart, and a way to make purchases. The store items would be kept in some kind of database.

II. Cài đặt

Cơ bản

Bạn sẽ cần JRE (Java Runtime Environment) phiên bản 1.6 trở lên. Có thể tải tại đây :

<http://java.sun.com/javase/downloads>

Cài đặt Solr

Solr hiện tại có thể tìm thấy ở đây <http://lucene.apache.org/solr>

Nếu hệ điều hành của bạn là Linux/Unix/OSX thì tải file có đuôi là .gzip. Còn hệ điều hành Windows thì tải file có đuôi là .zip

Giải thích đôi chút về Solr : Solr chạy bên trong một Java servlet container như Tomcat, Jetty hay Resin. Các bước để cài đặt như sau

1. Giải nén Solr đến thư mục mong muốn của bạn.
2. Mở folder sau : ...`\solr-4.4.x\example\`

3. Chạy bằng cmd : `cd ... \solr-4.4.x\example\`
4. Chạy lệnh : `java -jar start.jar`
Lệnh này sẽ khởi động Solr trên Jetty.

III. Chạy Solr

Sau khi đã cài đặt như các bước trên, hãy truy cập vào Solr như sau:

<http://hostname:8983/solr> (thường dùng nhất là <http://localhost:8983/solr>)

Nếu sau đó trình duyệt hiện ra tương tự thế này tức là bạn đã cài đặt và chạy thành công:

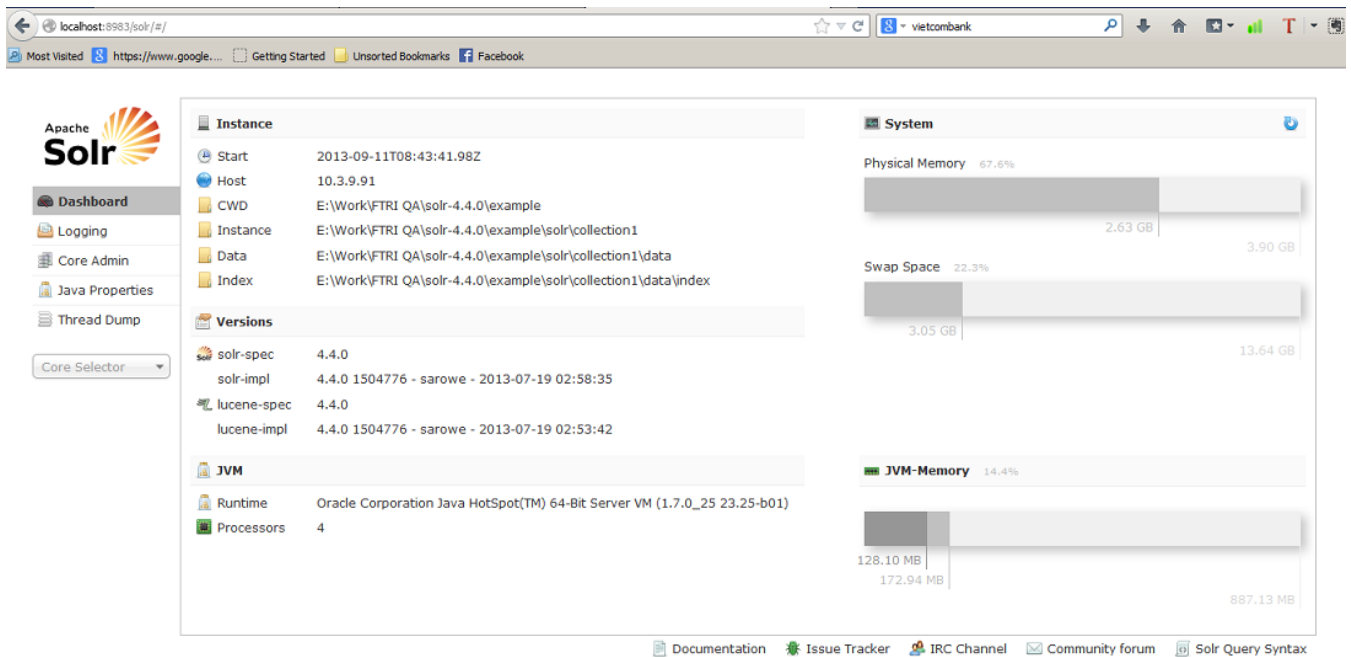


Figure 1 Giao diện quản lý Solr

IV. Configuration

Có 2 file config chính là: **schema.xml** và **solrconfig.xml**

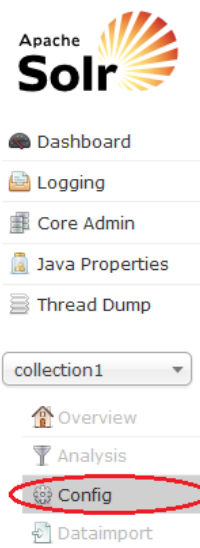
Schema.xml dùng để config các field (trường) dữ liệu, cách lưu trữ dữ liệu để sau này ta lưu data vào đó.

Solrconfig.xml dùng để config các chức năng (cụ thể ở đây là các request handlers và các tính năng khác)

2 file này nằm trong thư mục `solr-4.4.x\example\solr\collection1\conf`

Có thể truy cập file Solrconfig.xml bằng cách vào Admin của Solr (view-only) :

<http://localhost:8983/solr/#/collection1/config>



```
http://localhost:8983/solr/collection1/admin/file?file=solrconfig.xml&contentType=text/xml; charset=utf-8

<?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.
-->
```

Chọn collection1 -> Config thì sẽ xem được file Solrconfig.xml trực tiếp

Cách thêm trường mới

Để thêm trường mới (parameter/field/schema v.v...), tìm file Schema.xml và mở notepad++ để edit như sau

```
130 -->
131 <field name="id" type="uuid" indexed="true" stored="true" default="NEW" required="true"/>
132 <field name="title" type="text_general" indexed="true" stored="true" multiValued="true"/>
133 <field name="subject" type="text_general" indexed="true" stored="true"/>
134 <field name="description" type="text_general" indexed="true" stored="true"/>
135 <field name="comments" type="text_general" indexed="true" stored="true"/>
136 <field name="author" type="text_general" indexed="true" stored="true"/>
137 <field name="keywords" type="text_general" indexed="true" stored="true"/>
138 <field name="category" type="text_general" indexed="true" stored="true"/>
139 <field name="resourceName" type="text_general" indexed="true" stored="true"/>
140 <field name="url" type="text_general" indexed="true" stored="true"/>
141 <field name="content_type" type="string" indexed="true" stored="true" multiValued="true"/>
142 <field name="last_modified" type="date" indexed="true" stored="true"/>
143 <field name="links" type="string" indexed="true" stored="true" multiValued="true"/>
144 <field name="answer" type="text_general" indexed="true" stored="true" multiValued="true"/>
145 <field name="question" type="text_general" indexed="true" stored="true" multiValued="true"/>
146 <field name="type1" type="text_general" indexed="true" stored="true" multiValued="true"/>
147 <field name="type2" type="text_general" indexed="true" stored="true" multiValued="true"/>
148 <field name="count" type="text_general" indexed="true" stored="true" multiValued="true"/>
149 <field name="test_prefix" type="string" indexed="true" stored="false" multiValued="true"/>
```

Ví dụ ta muốn thêm một trường là « answer » vào Schema.xml để sau này thêm mới các data vào trường đó -> index ; thì phải thêm dòng này, ví dụ

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

Các phần như “type”, “indexed”, “stored” và “multiValued” là các attribute (thuộc tính) của trường đó. Và 1 trường có các thuộc tính như sau:

- `default`
 - Đây là giá trị mặc định cho trường này nếu không đc gán khi add document (sẽ giải thích phần add document ở mục sau)
- `indexed=true|false`
 - “True” nếu trường này cần được đánh chỉ mục (indexed). Một trường có thể tìm được (searchable), sắp xếp được (sortable) và chỉnh sửa (facetable) nếu và chỉ nếu nó được indexed (thật ra cũng không rõ cái facetable nghĩa là gì vì chưa thử và cũng không có nhu cầu thử).
- `stored=true|false`
 - “True” nếu trường này cần được lưu vĩnh viễn trong phần data của Solr. Nếu là “False” thì lần sau khởi động sẽ phải add lại và index lại.
- `compressed=true|false`
 - “True” nếu trường này cần được lưu trữ bằng dạng nén Gzip (áp dụng cho các kiểu trường nén được (compressible))
- `compressThreshold=<integer>`
- `multiValued=true|false`
 - “True” nếu trường này có thể chứa nhiều giá trị cho mỗi document. Ví dụ: nó có thể xuất hiện nhiều lần trong một document.
- `omitNorms=true|false`
 - Đây có thể được coi là một option nâng cao.
 - Để là “true” để bỏ qua việc chuẩn hoá với trường này (sẽ vô hiệu hoá việc chuẩn hoá độ dài và làm tăng thời gian index cho trường này, do đó sẽ tiết kiệm đc bộ nhớ). Chỉ có các trường full-text hoặc các trường cần tăng thời gian index mới cần chuẩn hoá.
- `type`
 - Loại lưu trữ. Có rất nhiều loại lưu trữ như: date, binary, string, text, text_general, uuid ... tùy thuộc vào mục đích của chúng ta khi sử dụng.
 - Ta có thể thêm/chỉnh sửa phần type này bằng cách thêm thẻ <fieldtype> như sau

`<fieldType name="pfloat" class="solr.FloatField"/>`

Lưu ý: Loại lưu trữ rất ảnh hưởng đến thuật toán tìm kiếm cũng như các chức năng khác. Ví dụ, loại “**text_general**” sẽ lưu dạng text, nhưng các từ trong này sẽ được tokenized (tách ra) riêng biệt (phù hợp với các module như *spellchecker* v.v...), nhưng loại “**string**” sẽ lưu nguyên văn cả câu/đoạn mà không tách bất cứ câu nào ra. Loại này sẽ phù hợp với các module như *prefix suggestion/sentence suggestion*.

Ta có thể tự định nghĩa một loại lưu trữ khác, ví dụ tiếng Việt chẳng hạn (**text_vi**) tuy nhiên ta phải code và embed class tiếng Việt này cũng như file stopwords và các modules tokenized vào Solr. *Việc này cần thời gian nghiên cứu thêm vì hiện tại 1-2 cá nhân rất khó có thể thêm ngôn ngữ mới vào Solr*, trừ phi có nguyên 1 đội nghiên cứu và phát triển. Ví dụ sẽ có dạng như sau:

```

<!-- Vietnamese -->
<!--
<fieldType name="text_vi" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer class="org.ftr.qa.analyzer.VNTokenizer"/>

    <filter class="org.ftr.qa.analyzer.VNStopWordsProvider" ignoreCase="true" words="lang/stopwords_vi.txt"/>

    <filter class="org.ftr.qa.analyzer.VNStemFilterFactory"/>
  </analyzer>
</fieldType>
-->

```

Hình ảnh chỉ mang tính chất minh họa. Đại khái sẽ có dạng như thế, nhưng phần filter class sẽ phải code riêng ra

Cách thêm Request Handler

Request Handler là một trong những function mạnh nhất và phổ biến nhất hiện nay. Một trong những request handler mặc định mà Solr đang cung cấp cho chúng ta là: SearchHandler, Spellchecker, Suggestion, Update, v.v... (Tất nhiên sẽ phải config thêm)

- Search Handler: <http://wiki.apache.org/solr/SearchHandler> (truy cập vào đây để đọc hướng dẫn config search handler)

```

794 -->
795 <!-- SearchHandler
796
797 http://wiki.apache.org/solr/SearchHandler
798
799 For processing Search Queries, the primary Request Handler
800 provided with Solr is "SearchHandler" It delegates to a sequent
801 of SearchComponents (see below) and supports distributed
802 queries across multiple shards
803 -->
804 <requestHandler name="/select" class="solr.SearchHandler">
805 <!-- default values for query parameters can be specified, these
806 will be overridden by parameters in the request
807 -->
808
809
810 <lst name="defaults">
811 <str name="echoParams">explicit</str>
812 <str name="pf">question^100</str>
813 <str name="qf">question</str>
814 <int name="rows">10</int>
815 <str name="df">text</str>
816 </lst>
817

```

Đây là nội dung trong file Solrconfig.xml thể hiện phần config Search Handler

Mặc định thì SearchHandler của Solr chạy khá mượt, tức là có thể dùng ngay không cần edit trong file **Solrconfig.xml** làm gì. Tuy nhiên nếu để tăng performance hoặc customize thì có thể chỉnh ở đây. Ví dụ ở ảnh trên tác giả đã boost việc ưu tiên tìm kiếm cho trường question lên mũ 100 lần (so với các trường khác).

- Update Request Handler

```

1010  <!--
1011  <requestHandler name="/update" class="solr.UpdateRequestHandler">
1012  <!-- See below for information on defining
1013  updateRequestProcessorChains that can be used by name
1014  on each Update Request
1015  -->
1016  <!--
1017  <lst name="defaults">
1018  <str name="update.chain">dedupe</str>
1019  </lst>
1020  -->
1021  </requestHandler>

```

Đây là nội dung trong file *Solrconfig.xml* thể hiện phần config Update Request Handler

Tương tự như Search Handler thì Solr đã cung cấp chức năng này khá tốt, hầu như không phải tùy chỉnh gì.

- Spell Checker: <http://wiki.apache.org/solr/SpellCheckComponent>

Với module Spellchecker này thì ta phải config 1 chút. Truy cập vào link trên, ta sẽ thấy có nhiều cách để check phần chính tả (spellchecker).

```

1243  <!-- Spell Check
1244
1245  The spell check component can return a list of alternative spelling
1246  suggestions.
1247
1248  http://wiki.apache.org/solr/SpellCheckComponent
1249  -->
1250  <searchComponent name="spellcheck" class="solr.SpellCheckComponent">
1251
1252  <str name="queryAnalyzerFieldType">text_general</str>
1253
1254  <!-- Multiple "Spell Checkers" can be declared and used by this
1255  component
1256  -->
1257
1258  <!-- a spellchecker built from a field of the main index -->
1259  <lst name="spellchecker">
1260  <str name="name">default</str>
1261  <str name="field">text</str>
1262  <str name="classname">solr.DirectSolrSpellChecker</str>
1263  <!-- the spellcheck distance measure used, the default is the internal levenshtein -->
1264  <str name="distanceMeasure">internal</str>
1265  <!-- minimum accuracy needed to be considered a valid spellcheck suggestion -->
1266  <float name="accuracy">0.5</float>
1267  <!-- the maximum #edits we consider when enumerating terms: can be 1 or 2 -->
1268  <int name="maxEdits">2</int>
1269  <!-- the minimum shared prefix when enumerating terms -->
1270  <int name="minPrefix">1</int>
1271  <!-- maximum number of inspections per result. -->
1272  <int name="maxInspections">5</int>
1273  <!-- minimum length of a query term to be considered for correction -->
1274  <int name="minQueryLength">4</int>
1275  <!-- maximum threshold of documents a query term can appear to be considered for correction -->
1276  <float name="maxQueryFrequency">0.01</float>
1277  <!-- uncomment this to require suggestions to occur in 1% of the documents
1278  <float name="thresholdTokenFrequency">.01</float>
1279  -->
1280  </lst>
1281

```



```

1282 <!-- a spellchecker that can break or combine words. See "/spell" handler below for usage -
1283 <lst name="spellchecker">
1284   <str name="name">wordbreak</str>
1285   <str name="classname">solr.WordBreakSolrSpellChecker</str>
1286   <str name="field">name</str>
1287   <str name="combineWords">true</str>
1288   <str name="breakWords">true</str>
1289   <int name="maxChanges">10</int>
1290 </lst>
1291
1292 <!-- a spellchecker that uses a different distance measure -->
1293 <!--
1294   <lst name="spellchecker">
1295     <str name="name">jarowinkler</str>
1296     <str name="field">spell</str>
1297     <str name="classname">solr.DirectSolrSpellChecker</str>
1298     <str name="distanceMeasure">
1299       org.apache.lucene.search.spell.JaroWinklerDistance
1300     </str>
1301   </lst>
1302   -->
1303
1304 <!-- a spellchecker that use an alternate comparator
1305
1306   comparatorClass be one of:
1307     1. score (default)
1308     2. freq (Frequency first, then score)
1309     3. A fully qualified class name
1310   -->
1311 <!--
1312   <lst name="spellchecker">
1313     <str name="name">freq</str>
1314     <str name="field">lowerfilt</str>
1315     <str name="classname">solr.DirectSolrSpellChecker</str>
1316     <str name="comparatorClass">freq</str>
1317   </lst>
1318   -->

```

Đây là phần config cho module Spellchecker trong Solrconfig.xml. Nhìn qua ta có thể thấy Spellchecker có thể sử dụng rất nhiều thuật toán khác nhau (để sử dụng chỉ cần bỏ dấu quote). Có thuật toán thì sử dụng distance (như kiểu Levenshtein), có thuật toán sử dụng Word Break, có thuật toán sử dụng ngữ liệu kiểu file-based spell checker, v.v...

V. Thêm document vào Solr

Ở đây tôi giới thiệu 3 cách thêm document vào Solr (sau khi thêm document thì Solr sẽ tự động index cho chúng ta)

Chạy bằng cmd

1. Truy cập vào thư mục *example/exampledocs* trên cmd
2. Chạy lệnh sau

```
java -Ddata=files -jar post.jar *.xml
```

Lệnh này sẽ add tất cả các file có đuôi *.xml* trong thư mục đó để add vào server

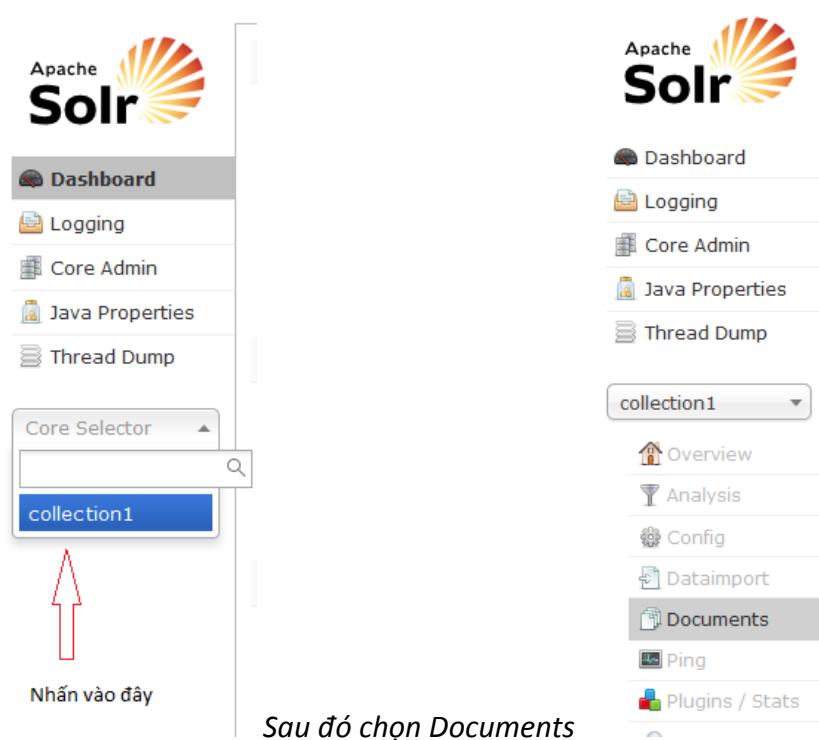
3. Commit lên server bằng cách thay tham số sau **java** bằng


```
-Durl=http://localhost:8983/solr/update
```

```
-Dcommit=yes
```

Chạy bằng Solr admin

1. Chọn **collector1** (hoặc bất cứ collector nào bạn tự tạo ra)



2. Lúc này màn hình hiện ra như sau, bạn hãy chọn **XML** ở dưới phần **Document Type**. Và copy các trường bạn muốn index vào phần **Document(s)** như sau

Request-Handler (qt) /update

Document Type XML

Document(s)

```
<doc>
  <field name="id">MA147LL/A</field>
  <field name="name">Apple 60 GB iPod with Video Playback
  Black</field>
  <field name="manu">Apple Computer Inc.</field></doc>
```

Commit Within 1000

Overwrite true

Submit Document

Status: success

Response:

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 53
  }
}
```

3. Ấn Submit. Lúc này khi thành công sẽ báo lại là success.

Lưu ý: Format của phần document này sẽ có dạng như sau

```

<doc>
  <field name="id">EN7800GTX/2DHTV/256M</field>
  <field name="name">ASUS Extreme N7800GTX/2DHTV (256 MB)</field>
  <!-- Denormalized -->
  <field name="manu">ASUS Computer Inc.</field>
  <!-- Join -->
  <field name="manu_id_s">asus</field>
  <field name="cat">electronics</field>
  <field name="cat">graphics card</field>
  <field name="features">NVIDIA GeForce 7800 GTX GPU/VPU clocked at 486MHz</field>
  <field name="features">256MB GDDR3 Memory clocked at 1.35GHz</field>
  <field name="features">PCI Express x16</field>
  <field name="features">Dual DVI connectors, HDTV out, video input</field>
  <field name="features">OpenGL 2.0, DirectX 9.0</field>
  <field name="weight">16</field>
  <field name="price">479.95</field>
  <field name="popularity">7</field>
  <field name="store">40.7143,-74.006</field>
  <field name="inStock">>false</field>
  <field name="manufacturedate_dt">2006-02-13T15:26:37Z/DAY</field>
</doc>

```

(Thao khảo example/exampledocs)

Ngoài ra chúng ta còn có thể upload file có sẵn bằng cách chọn các file .xml có định dạng tương tự như trên và up lên Solr

Chỉ cần ấn Browse và chọn file bạn cần upload

Chạy bằng lệnh trên Eclipse/Netbeans

1. Đoạn code sau mô tả việc add các trường vào document

```

SolrInputDocument doc = new SolrInputDocument();
doc2.addField( "id", "B0011");
doc2.addField( "name", "SolrConfig");
doc2.addField( "price", 20 );

```

(Trong đó, các trường id, name và price phải được khai báo trong file schema.xml)

2. Tạo Collection cho SolrDoc

```
Collection<SolrInputDocument> docs = new ArrayList<SolrInputDocument>();  
docs.add( doc1 );  
docs.add( doc2 );
```

3. Add document đó lên Solr

```
server.add( docs );
```

4. Cuối cùng, commit lên server

```
server.commit();
```

Lưu ý: Có thể không cần add trường id bằng tay vì ta có thể tùy chỉnh cho trường id là tự động (sẽ có hướng dẫn cụ thể ở mục sau)

VI. Xóa dữ liệu/document trên Solr Admin

Lặp lại các bước như phần add document ở mục V. Chỉ phần **Document Type** thành **XML** và paste đoạn code sau

```
<delete><query>*:*/</query></delete><commit/>
```

Vào phần **Document(s)**

Request-Handler (qt)
/update

Document Type
XML

Document(s)
<delete><query>*:*/</query></delete><commit/>

Commit Within
1000

Overwrite
true

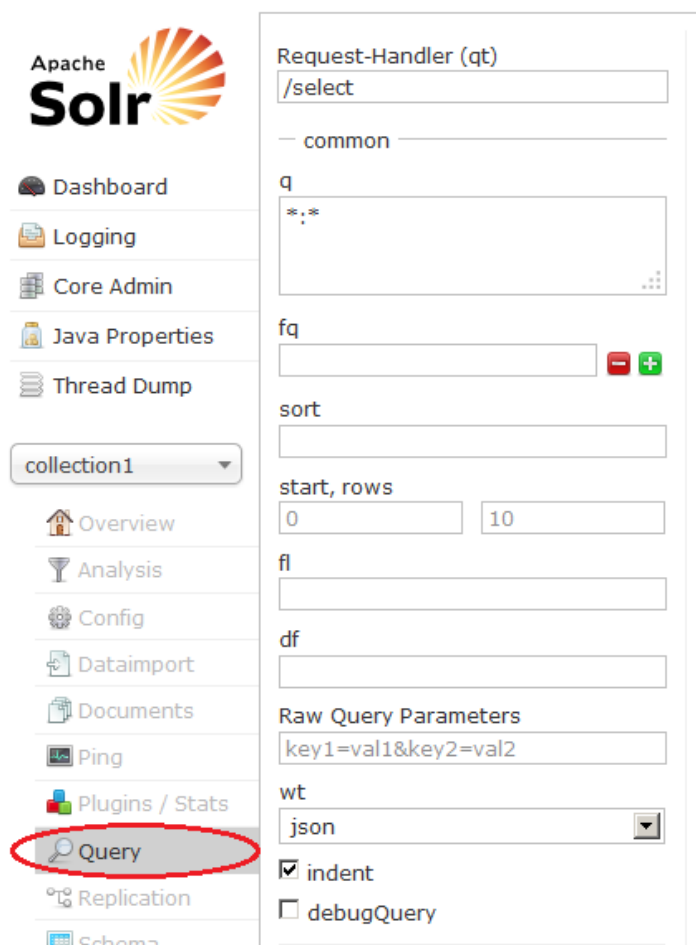
Submit Document

Status: success
Response:
{
 "responseHeader": {
 "status": 0,
 "QTime": 524
 }
}

VII. Tìm kiếm trên Solr Admin

Truy cập phần tìm kiếm như sau:

1. Chọn collection1 -> Query



The screenshot displays the Apache Solr Admin web interface. On the left, a sidebar menu lists various administrative functions: Dashboard, Logging, Core Admin, Java Properties, Thread Dump, a dropdown menu currently showing 'collection1', Overview, Analysis, Config, Dataimport, Documents, Ping, Plugins / Stats, **Query** (highlighted with a red circle), Replication, and Schema. The main content area on the right is titled 'Request-Handler (qt)' and contains several input fields and checkboxes. The 'Request-Handler (qt)' field is set to '/select'. Below it, a 'common' section contains a 'q' field with the value '*:*'. The 'fq' field is empty, accompanied by minus and plus icons. The 'sort' field is empty. The 'start, rows' section has 'start' set to 0 and 'rows' set to 10. The 'fl' and 'df' fields are empty. The 'Raw Query Parameters' field contains 'key1=val1&key2=val2'. The 'wt' field is set to 'json' via a dropdown menu. At the bottom, there are checkboxes for 'indent' (checked) and 'debugQuery' (unchecked).

Chọn Query như hình

2. Sau đó chạy query như sau:

The screenshot shows the Solr Request-Handler (qt) interface. On the left, the 'q' field is populated with 'question:FPT,là,gì', which is circled in red. Other fields like 'fq', 'sort', 'start, rows', 'fl', 'df', 'Raw Query Parameters', 'wt', and checkboxes for 'indent', 'debugQuery', and 'dismax' are also visible. On the right, the JSON response is displayed, showing the query parameters and the search results.

```
http://localhost:8983/solr/collection1/select?q=question%

{
  "responseHeader": {
    "status": 0,
    "QTime": 46,
    "params": {
      "indent": "true",
      "q": "question:FPT,là,gì",
      "_": "1379493393935",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 1053,
    "start": 0,
    "docs": [
      {
        "type1": [
          "NET"
        ],
        "type2": [
          "URL"
        ],
        "count": [
          "S"
        ],
        "question": [
          "FPT là gì? "
        ]
      }
    ]
  }
}
```

Lưu ý: dấu “,” là để ngăn cách giữa các từ (ta có thể hiểu là dấu cách). Nếu ta gõ dấu cách thì query sẽ bị lỗi vì ở đây chúng ta đang sử dụng với góc nhìn là developer. Ví dụ như lên google khi ta tìm từ “FPT là gì” thì trên URL nó sẽ là “FPT+là+gì”. Còn ở đây thì dấu “,” sẽ có vai trò tương tự như dấu “+” trên Google.

Format của phần “q” sẽ là dạng *: * với * ở bên trái là tên trường (ví dụ ở đây là **question**), * bên phải sẽ là giá trị (ví dụ ở đây là “FPT,là,gì”). Nếu để default thì nó sẽ ra toàn bộ. Tương tự nếu cần tìm cho trường **answer** thì sẽ phải là “answer:Chủ,tịch,FPT”. Nếu muốn tìm cả 2 trường question và answer thì sẽ là “question:FPT,là,gì AND answer:FPT,là”

Lưu ý lần 2: Đây chỉ là một search engine chứ ko phải máy giải đáp nên máy sẽ trả lại phần text/document nào liên quan nhất.

3. Các parameters: <http://wiki.apache.org/solr/CommonQueryParameters>

- **df:** default field. Nếu khai báo ở đây nó sẽ overwrite trường mặc định trong Schema.xml
- **fl:** field list. Một option để hiển thị kết quả theo field list này (mặc định là từ lớn đến bé). Ví dụ ta muốn hiển thị các kết quả tìm kiếm theo score từ cao đến thấp, ta sử dụng: “fl:*,score”.

Request-Handler (qt)

/select

common

q

question:FPT là gì AND
answer:FPT là

fq

sort

start, rows

0 10

fl

*,score

df

http://localhost:8983/solr/collection1/select?q=question%3AFPT%20là%20gì%20AND%20answer%3AFPT%20là%20gì

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 37,
    "params": {
      "fl": "*,score",
      "indent": "true",
      "q": "question:FPT là gì AND answer:FPT là",
      "_": "1379493853047",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 969,
    "start": 0,
    "maxScore": 1.3211255,
    "docs": [
```

- **fq**: filter query. Parameter này được sử dụng để xác định một truy vấn (query) mà có thể giới hạn tập hợp kết quả/documents được trả lại mà *không làm ảnh hưởng đến score*. Ví dụ: *fq=popularity:[10 TO *]* dùng để trả về các tập kết quả/documents có độ phổ biến cao hơn 10.

Lưu ý: Có thể dùng nhiều lần tham số này bằng cách dùng toán tử "&". Ví dụ *fq=popularity:[10 TO *]&fq=section:0* hoặc toán tử "+" như *fq=+popularity:[10 TO *]+section:0*.

Mọi thắc mắc xin liên hệ:

Email: contact@linhkid.com

Skype: linhkid91