



# **Painless Full Text Searching with SOLR**

**Michael Kimsal**

WebDevRadio.com

# Outline

---

- What is SOLR?
- Why use SOLR?
- Installation
- Run through with sample data
- Lucene syntax
- SOLR advanced query control panel
- Create new schema
- Create example application
- Questions?

# What is SOLR?

---

- SOLR is a REST layer for Lucene
- Began life at CNET to provide a robust search system
- Joined Apache Incubator in January 2006
- Graduated to Lucene sub-project status in January 2007

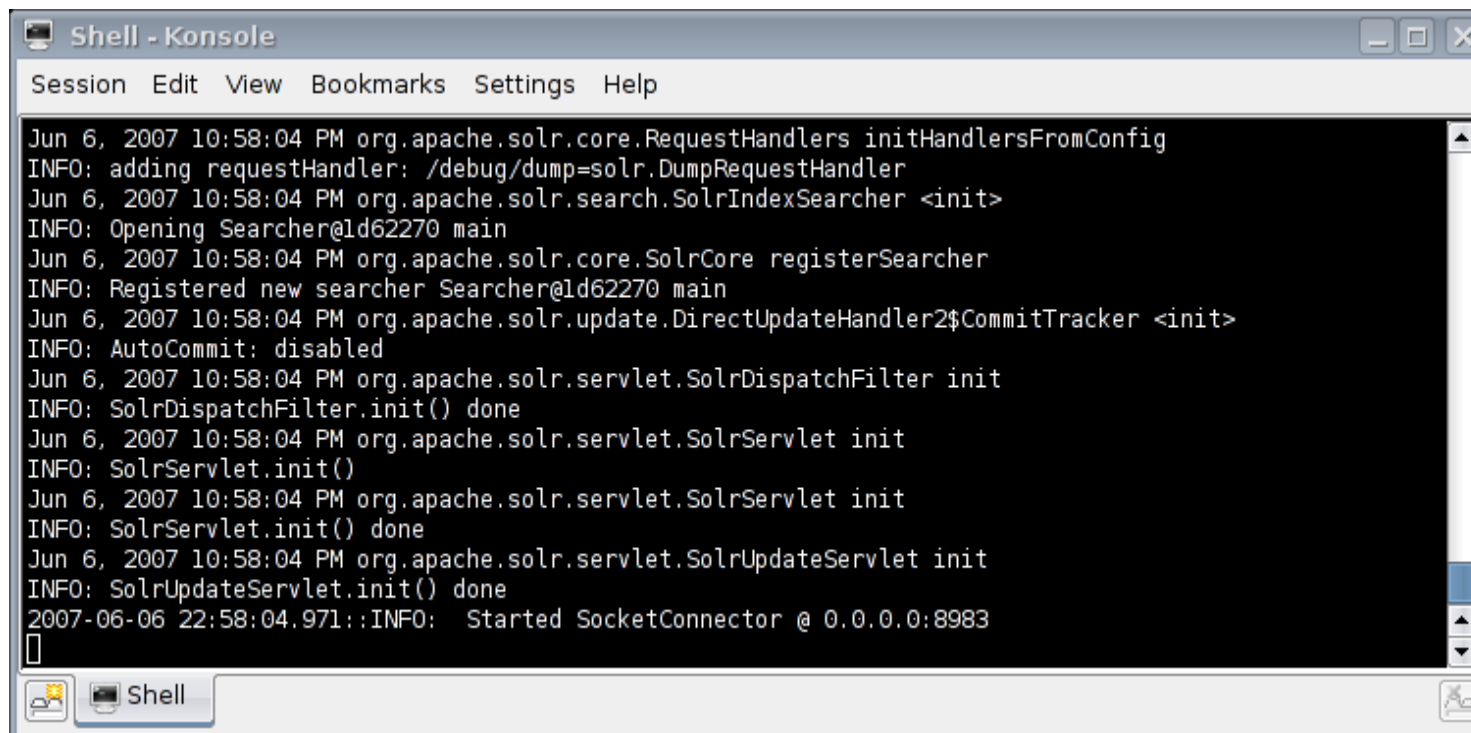
# Why use SOLR?

---

- Easy to set up and get started
- Powerful full text searching
- Cross platform - Java and REST
- Under active development
- Fast
- Adds extra functionality on top of Lucene:
  - replication
  - CSV importing
  - JSON results
  - results highlighting
  - synonym support

# Installation

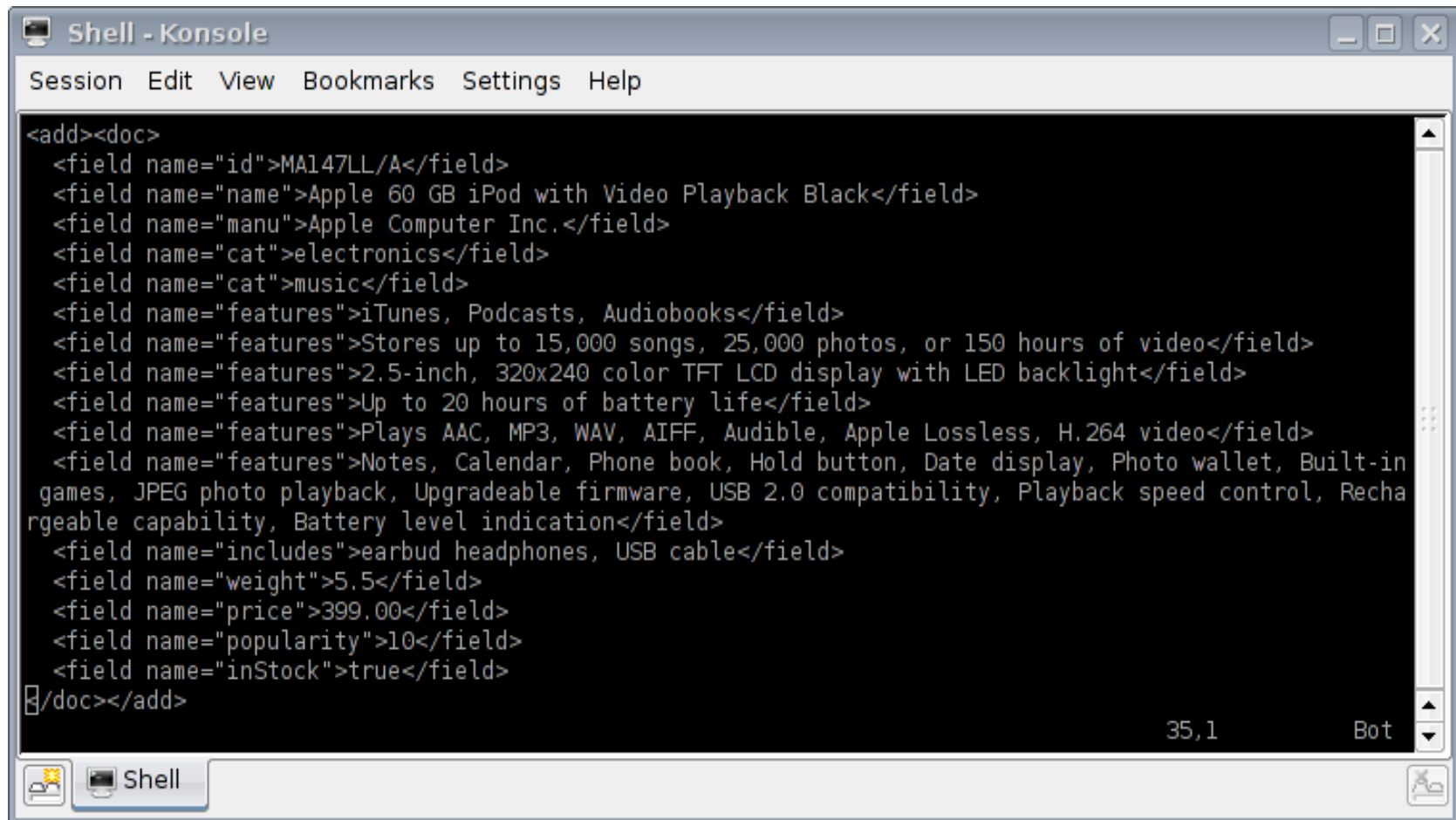
- Quick download
- Untar, jump to example directory
- `java -jar ./start.jar`
- Bundled with Jetty webserver on port 8983



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
Jun 6, 2007 10:58:04 PM org.apache.solr.core.RequestHandlers initHandlersFromConfig
INFO: adding requestHandler: /debug/dump=solr.DumpRequestHandler
Jun 6, 2007 10:58:04 PM org.apache.solr.search.SolrIndexSearcher <init>
INFO: Opening Searcher@1d62270 main
Jun 6, 2007 10:58:04 PM org.apache.solr.core.SolrCore registerSearcher
INFO: Registered new searcher Searcher@1d62270 main
Jun 6, 2007 10:58:04 PM org.apache.solr.update.DirectUpdateHandler2$CommitTracker <init>
INFO: AutoCommit: disabled
Jun 6, 2007 10:58:04 PM org.apache.solr.servlet.SolrDispatchFilter init
INFO: SolrDispatchFilter.init() done
Jun 6, 2007 10:58:04 PM org.apache.solr.servlet.SolrServlet init
INFO: SolrServlet.init()
Jun 6, 2007 10:58:04 PM org.apache.solr.servlet.SolrServlet init
INFO: SolrServlet.init() done
Jun 6, 2007 10:58:04 PM org.apache.solr.servlet.SolrUpdateServlet init
INFO: SolrUpdateServlet.init() done
2007-06-06 22:58:04.971::INFO: Started SocketConnector @ 0.0.0.0:8983
```

# Run through with sample data

- Sample schema is a product system
- ipods, cameras, monitors, etc.



```
Shell - Konsole
Session Edit View Bookmarks Settings Help

<add><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>
  <field name="cat">electronics</field>
  <field name="cat">music</field>
  <field name="features">iTunes, Podcasts, Audiobooks</field>
  <field name="features">Stores up to 15,000 songs, 25,000 photos, or 150 hours of video</field>
  <field name="features">2.5-inch, 320x240 color TFT LCD display with LED backlight</field>
  <field name="features">Up to 20 hours of battery life</field>
  <field name="features">Plays AAC, MP3, WAV, AIFF, Audible, Apple Lossless, H.264 video</field>
  <field name="features">Notes, Calendar, Phone book, Hold button, Date display, Photo wallet, Built-in
  games, JPEG photo playback, Upgradeable firmware, USB 2.0 compatibility, Playback speed control, Recha
  rgeable capability, Battery level indication</field>
  <field name="includes">earbud headphones, USB cable</field>
  <field name="weight">5.5</field>
  <field name="price">399.00</field>
  <field name="popularity">10</field>
  <field name="inStock">true</field>
</doc></add>
```

# Querying with control panel

---

- Lucene syntax overview
  - <http://lucene.apache.org/java/docs/queryparsersyntax.html>
- Full(er) control panel interface

# Lucene syntax

---

- Required search term – use a “+”
  - +ipod
  - +belkin
- Field-specific searching – use fieldName
  - name:ipod
  - manu:belkin
- Wildcard searching – use \* or ?
  - ip?d
  - belk\*
  - \*deo (currently requires modifying solr source)

## Lucene syntax part two

---

- Range searching
  - timestamp:[2006-07-16T12:30:00Z to \*]
    - Time needs to be full ISO
- Proximity searching – use a “~”
  - "video ipod"~3 – up to 3 words apart
- Fuzzy searches – use a “~”
  - ipod~ will find ipod and ipods
  - belkin~0.7 will find words close spellings

# Full control panel interface

---

- Start row/max rows – pagination
- Output type
  - standard (xml), python, json, ruby, xslt
- Enable highlighting
  - fields to highlight
  - does not currently work on wildcard matches
    - please vote for SOLR-218!
    - <https://issues.apache.org/jira/browse/SOLR-218>

# Create new schema

---

- Blog aggregator
- Will track feed URL, feed title, entry URL, entry title, entry body, author, publication date

```
<fields>
  <field name="id" type="string" indexed="true" stored="true"/>
  <field name="title" type="text" indexed="true" stored="true"/>
  <field name="description" type="text" indexed="true" stored="true"/>
  <field name="guid" type="text" indexed="false" stored="true"/>
  <field name="pubdate" type="date" indexed="true" stored="true"/>
  <field name="creator_author" type="text" indexed="true" stored="true"/>
  <field name="creator_link" type="text" indexed="true" stored="true"/>
  <field name="creator_email" type="text" indexed="true" stored="true"/>
  <field name="feed_title" type="text" indexed="true" stored="true"/>
  <field name="feed_url" type="text" indexed="false" stored="true"/>
  <!-- catchall field, containing all other searchable text fields (implemented
        via copyField further on in this schema -->
  <field name="text" type="text" indexed="true" stored="false" multiValued="true"/>
</fields>

<copyField source="title" dest="text"/>
<copyField source="description" dest="text"/>
<!-- field to use to determine and enforce document uniqueness. -->
<uniqueKey>id</uniqueKey>
```

# Create example application

---

- PHP5
- First part is the indexer program
- Read in INI file with list of blog URLs
- Write out XML file to run through CURL
- Two gotchas – entity handling
  - Escape all & to &amp;
  - Include entity reference file

# Example indexing application

```
<?php
include("simplepie/simplepie.inc");
$ini = parse_ini_file("blogs.ini", TRUE);
foreach($ini as $blogTitle=>$entry) {
    $nodes = array();
    $feed = new SimplePie($entry['FEED']);
    $items = $feed->get_items();
    echo "Pulling feed -> $blogTitle\n";
    $header = "<?xml version='1.0'?>\n".file_get_contents("./xhtml-lat1.ent")."\n";
    foreach($items as $key=>$item) {
        $author = $item->get_author(0);
        $node = "<field name=\"id\">".$item->get_permalink()."</field>\n";
        $node .= "<field name=\"title\">".entity_fix($item->get_title())."</field>\n";
        $node .= "<field name=\"description\">".entity_fix($item->get_description())."</field>\n";
        $node .= "<field name=\"pubdate\">".$item->get_date("c")."Z</field>\n";
        $node .= "<field name=\"creator_author\">".$author->get_name()."</field>\n";
        $node .= "<field name=\"creator_link\">".$author->get_link()."</field>\n";
        $node .= "<field name=\"creator_email\">".$author->get_email()."</field>\n";
        $node .= "<field name=\"feed_title\">".$title."</field>\n";
        $node .= "<field name=\"feed_url\">".$htmlUrl."</field>\n";
        $nodes[] = "<doc>\n$node</doc>";
    }
    $filename = "./output_".str_replace(" ", "_", $blogTitle).".xml";
    file_put_contents($filename, $header."<add>".implode("\n", $nodes)."</add>");
    $null = shell_exec("./silent_post.sh ".escapeshellcmd($filename)." > /dev/null & echo $!");
    unlink($filename);
}
function entity_fix($string) {
    $string = str_replace("&","&amp;","strip_tags($string));
    return $string;
}
?>
```

# Create search application

---

- Demonstrates JSON result type
- Sort syntax – sort=pubdate desc
  - multiple sort fields possible
- numFound property of response

# Example search application

```
<form method='get'>Search: <input type='text' name='query' /><input type='submit' /></form>
<?php
$perPage = 10;
$host = "http://localhost:8983/solr/select/?version=2.2&&indent=on&sort=pubdate+desc&wt=json&q=";
$query = urlencode(@$_GET['query']);
$start = (int)@$_GET['start'];

$solr = $host.$query."&rows=$perPage&sort=pubdate+desc&start=$start";
$json = json_decode(@file_get_contents($solr));
$totalRows = $json->response->numFound;

echo "Showing " . ($start+1) . " to " . ($start+$perPage) . " of " . (int)$totalRows . " results<br/>";
if ( ($start>0) && ($totalRows>$perPage) ) {
    echo "<a href='?query=$query&start=" . ($start-10) . "'>prev</a> | ";
} else {
    echo "prev | ";
}
if ($start<($totalRows-$perPage)) {
    echo "<a href='?query=$query&start=" . ($start+10) . "'>next</a> <hr/>";
} else {
    echo "next<hr/>";
}

if($json) {
    foreach($json->response->docs as $entry) {
        echo "Title: <a href='". $entry->guid ."' target='_blank'>". $entry->title . "</a><BR>";
        echo "Body: " . shortdesc($entry->description);
        echo "<hr/>";
    }
}

function shortdesc($string, $words = 20) {
    $parts = explode(" ", $string);
    $new = implode(" ", array_slice($parts, 0, $words)) . "...";
    return $new;
}
?>
```

# Resources

---

- <http://lucene.apache.org/solr>
- <http://wiki.apache.org/solr/SolrResources>
- <http://lucenebook.com>
- <http://del.icio.us/tag/solr>

# Questions?

---

Email: [mgkimsal@gmail.com](mailto:mgkimsal@gmail.com)

URL: <http://www.webdevradio.com>