INFORMATION RETRIEVAL

PROJECT 3

Team Glue

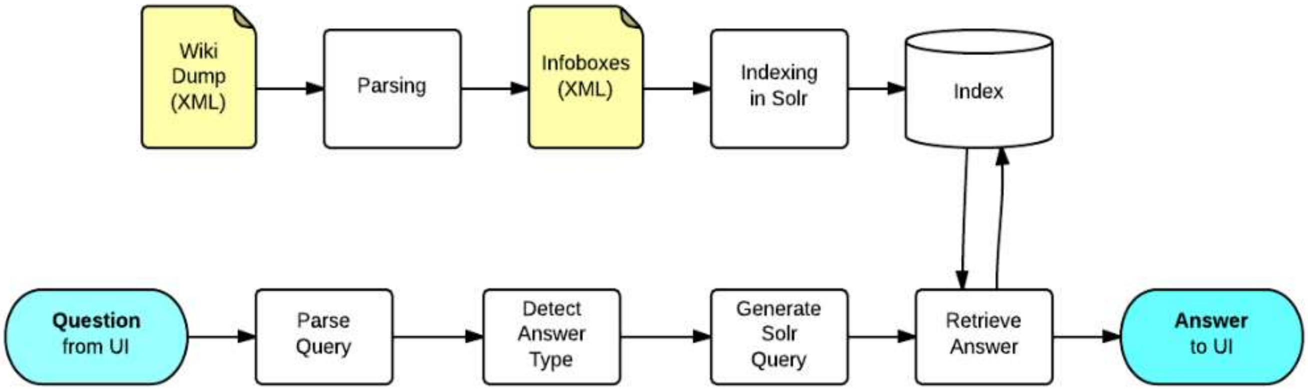
ANKIT JAIN

PRANAV GUPTA

DEEPAK VEERUPAPURAM

MILKY SAHU

SYSTEM DIAGRAM



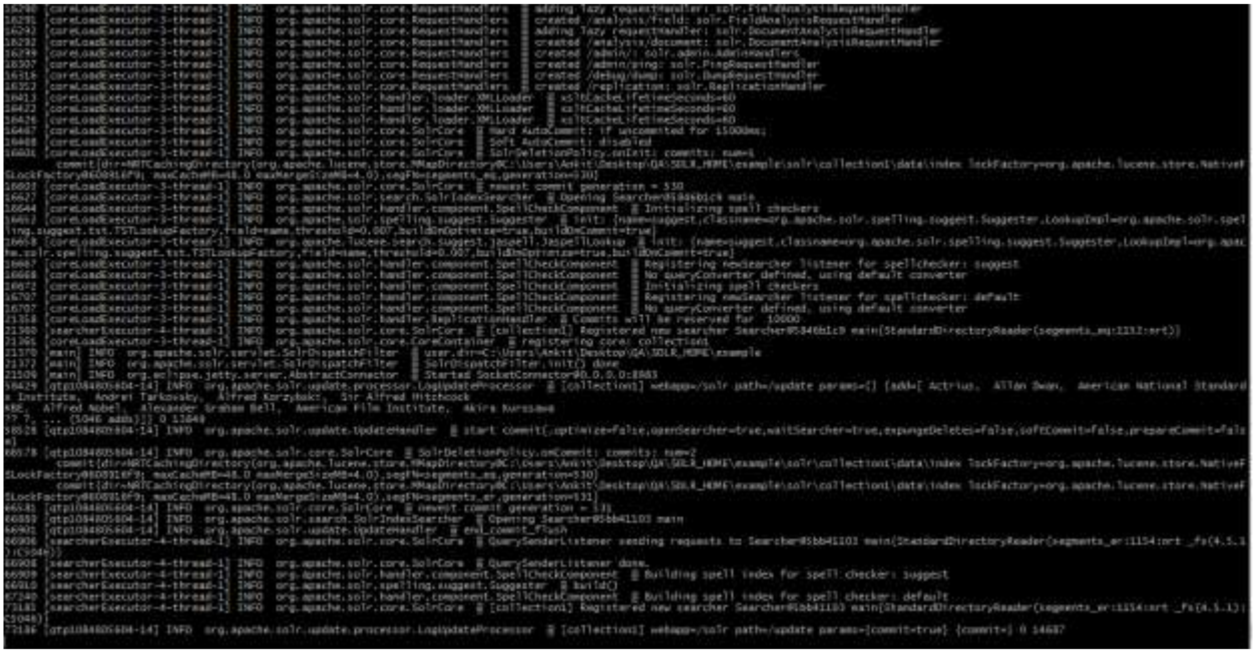
DESCRIPTION

- **Dump Processing & Parsing:**
  We start with downloading *enwiki* dump of Wikipedia documents which is an XML file. As the Question-Answer system is expected mainly to be based on Infobox section of each document, we parse the entire dump using SAX parser and extract all the Infoboxes in a specific Solr readable XML format, thereby reducing the size of the document to be handled.
  The format of the Infobox.xml is given as:

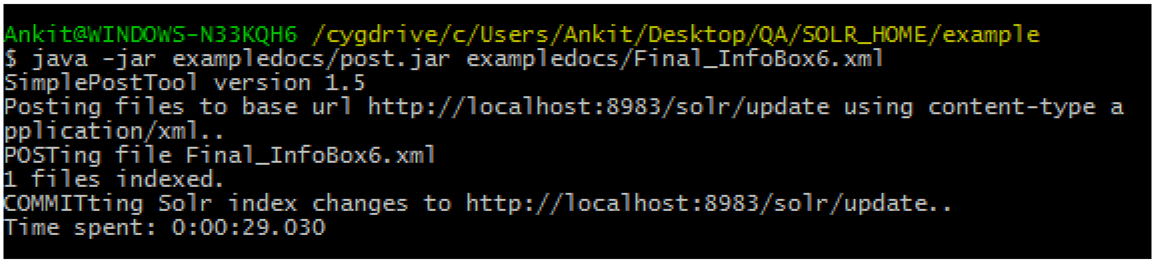
```
<add>
<doc>
<field name="name"><![CDATA[ Actrius]]></field>
<field name="alt"><![CDATA[ <!-- see WP:ALT -->]]></field>
<field name="caption"><![CDATA[ ]]></field>
<field name="director"><![CDATA[ [[Ventura Pons]]]]></field>
<field name="producer"><![CDATA[ Ventura Pons]]></field>
<field name="writer"><![CDATA[ Ventura Pons<br> [[Josep Maria Benet i Jornet]]]]></field>
<field name="starring"><![CDATA[ [[Núria Espert]]<br> [[Rosa Maria Sardà]]<br> [[Anna Lizaran]]<br>[[Mercè Pons]]]]></field>
<field name="music"><![CDATA[ [[Carles Cases]]]]></field>
<field name="cinematography"><![CDATA[ ]]></field>
<field name="editing"><![CDATA[ ]]></field>
<field name="studio"><![CDATA[ [[Els Films de la Rambla, S.A.]]]]></field>
<field name="distributor"><![CDATA[ ]]></field>
<field name="released"><![CDATA[ {{Film date|1996|||}}<!-- {{Film date|Year|Month|Day|Location}} -->]]></field>
<field name="runtime"><![CDATA[ 90 minutes]]></field>
<field name="country"><![CDATA[ Spain]]></field>
<field name="language"><![CDATA[ Catalan]]></field>
<field name="budget"><![CDATA[ ]]></field>
<field name="gross"><![CDATA[ ]]></field>
</doc> </add>
```

Here, the data enclosed within <doc></doc> tags is a single Infobox and there will be multiple Infoboxes in the file. <add></add> tags indicate the docs to be added to the index.

- **Indexing:**
  Schema.xml is configured according to the requirements (discussed later) and Infobox.xml is then placed in the SOLR_HOME/example/exampledocs folder to proceed with indexing.
  We start jetty by giving `java -jar start.jar` on command prompt

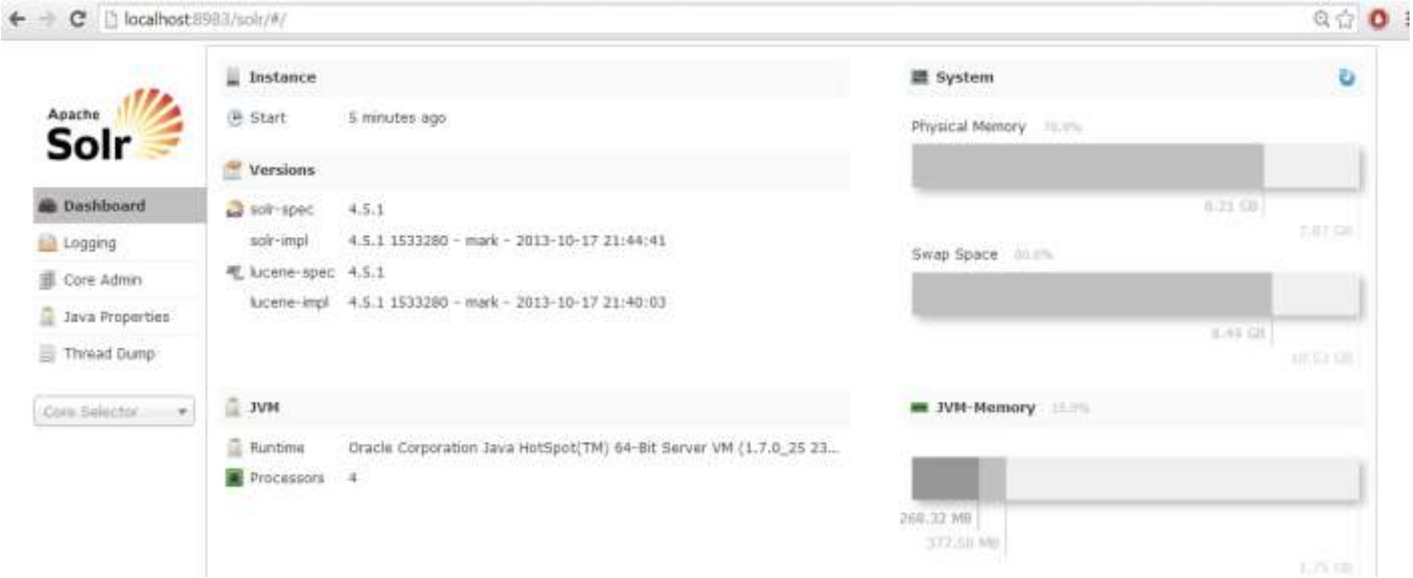


and post the file to Solr for indexing with the command `java -jar post.jar Infobox.xml.` If the indexing is successful we get the following success message:



```
Ankit@WINDOWS-N33KQH6 /cygdrive/c/Users/Ankit/Desktop/QA/SOLR_HOME/example
$ java -jar exampledocs/post.jar exampledocs/Final_InfoBox6.xml
SimplePostTool version 1.5
Posting files to base url http://localhost:8983/solr/update using content-type a
pplication/xml..
POSTing file Final_InfoBox6.xml
1 files indexed.
COMMITting Solr index changes to http://localhost:8983/solr/update..
Time spent: 0:00:29.030
```

The file has now been indexed and data has been stored in Solr index. The Solr Dashboard shows the following information after indexing:



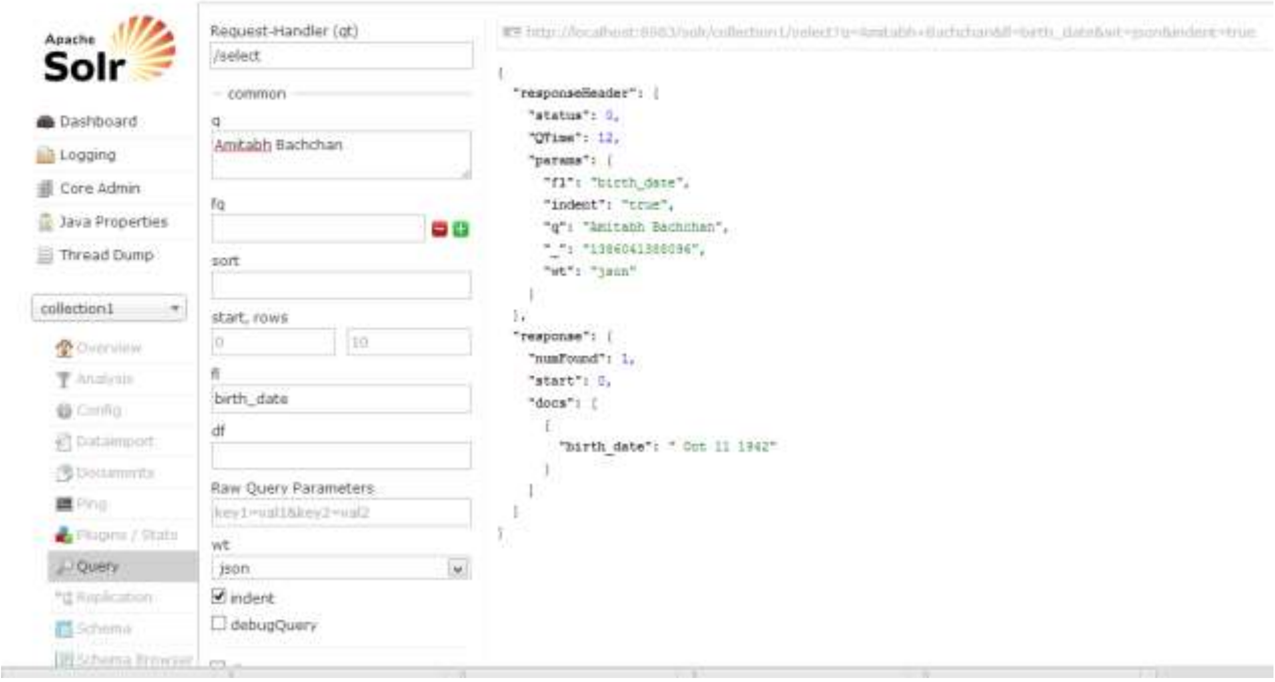
Start 5 minutes ago

solr-spec 4.5.1
solr-impl 4.5.1 1533280 - mark - 2013-10-17 21:44:41
lucene-spec 4.5.1
lucene-impl 4.5.1 1533280 - mark - 2013-10-17 21:40:03

Runtime Oracle Corporation Java HotSpot(TM) 64-Bit Server VM (1.7.0_25 23...
Processors 4

- **Query Processing:**
  We have extracted Infoboxes belonging to only 3 categories viz. Person, Organization and Film.
  Example Query
  Query of an XML search system for the question:
  When was Amitabh Bachchan born?
  In this question,
  When - question referring to the time period which in our context is a date
  was - stop word which we remove while processing the query
  Amitabh Bachchan - noun about which an Infobox is expected to be present in the knowledgebase
  born - verb which has to be extracted from the indexed data about Amitabh Bachchan.

  We have created a synonyms file for all the fields present in the Infoboxes. Also a hash map has been used which stores question+synonym as key and the Infobox field as value. Hence if the actual field name is birth_date, it will mapped to key value pair of {when+born, birth_date}. This will retrieve the birth_date field value from indexed data and return it to the Solr output. The output looks like this:



Request-Handler (qt): /select

q: Amitabh Bachchan

fq:

sort:

start, rows: 0 10

fl: birth_date

df:

Raw Query Parameters: key1=val1&key2=val2

wt: json

☑ indent

□ debugQuery

http://localhost:8983/solr/collection1/select?q=Amitabh+Bachchan&fl=birth_date&wt=json&indent=true

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 12,
    "params": {
      "fl": "birth_date",
      "indent": "true",
      "q": "Amitabh Bachchan",
      "_": "1386041388096",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 1,
    "start": 0,
    "docs": [
      {
        "birth_date": " Oct 11 1942"
      }
    ]
  }
}
```

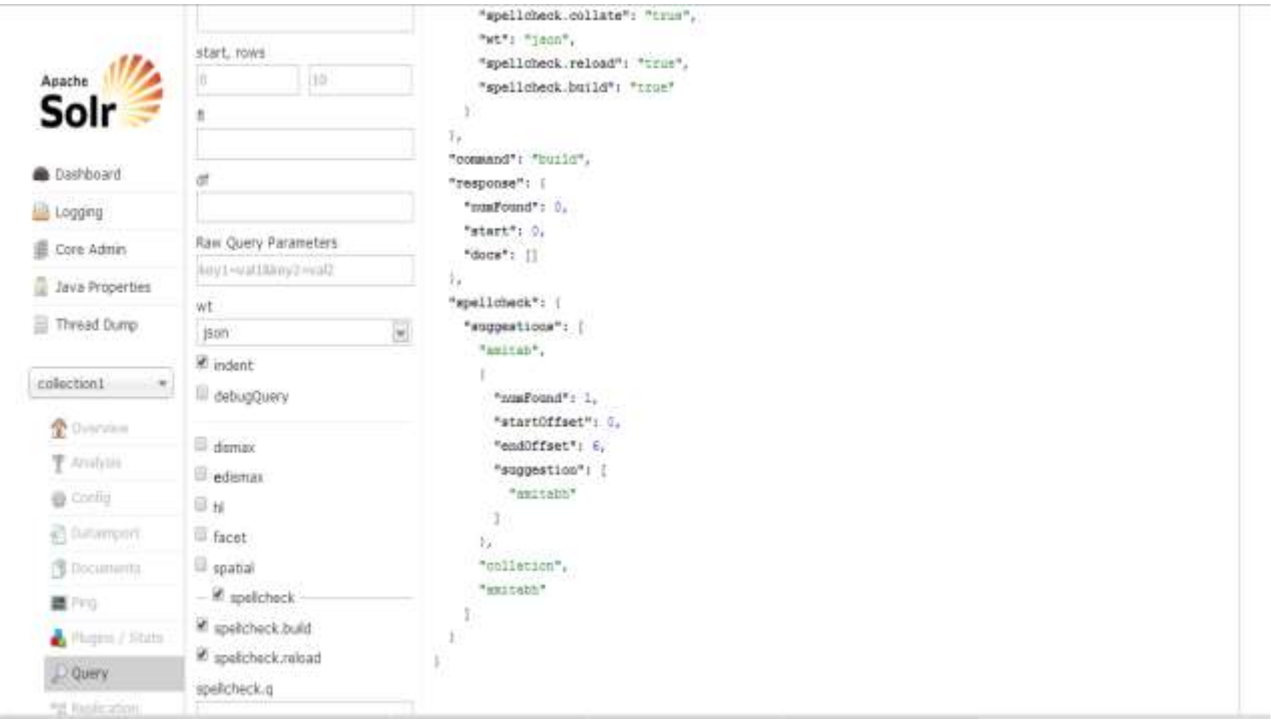
This output is directed to UI using the following approach.

- **Rendering Answer to UI:**
  Following components have been used to interface Solr to our User Interface:
  - Web Application based on asp.net
  - Web Service based on Java

  We take query input from user on web application based on asp.net i.e. the UI. The connection between UI and Solr is created using web service which we are maintaining on Java. User’s input is directed to Solr through web service. Solr server retrieves the result from Solr index and returns it back to web service which sends it to web application to display on UI.

SPECIAL FEATURES

- **Spell Check (Did you mean)**
  If a user will enter an incorrect spelling in the query, the UI will suggest the correct spelling by asking ‘did you mean’ to the user. The output on Solr UI looks like this:



start, rows: 0 10

fl

df

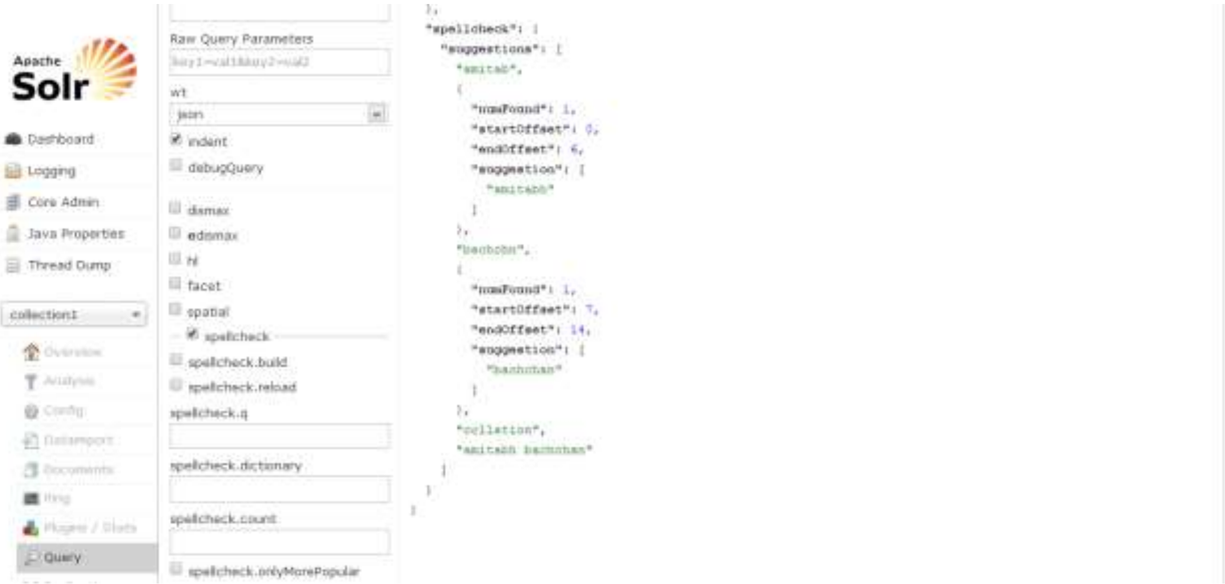
Raw Query Parameters: key1=val1&key2=val2

wt: json

☑ indent
□ debugQuery

□ dismax
□ edismax
□ hl
□ facet
□ spatial
☑ spellcheck
☑ spellcheck.build
☑ spellcheck.reload
spellcheck.q

```
      "spellcheck.collate": "true",
      "wt": "json",
      "spellcheck.reload": "true",
      "spellcheck.build": "true"
    }
  },
  "command": "build",
  "response": {
    "numFound": 0,
    "start": 0,
    "docs": []
  },
  "spellcheck": {
    "suggestions": [
      "amitab",
      {
        "numFound": 1,
        "startOffset": 0,
        "endOffset": 6,
        "suggestion": [
          "amitabh"
        ]
      },
      "collation",
      "amitabh"
    ]
  }
}
```



Raw Query Parameters: key1=val1&key2=val2

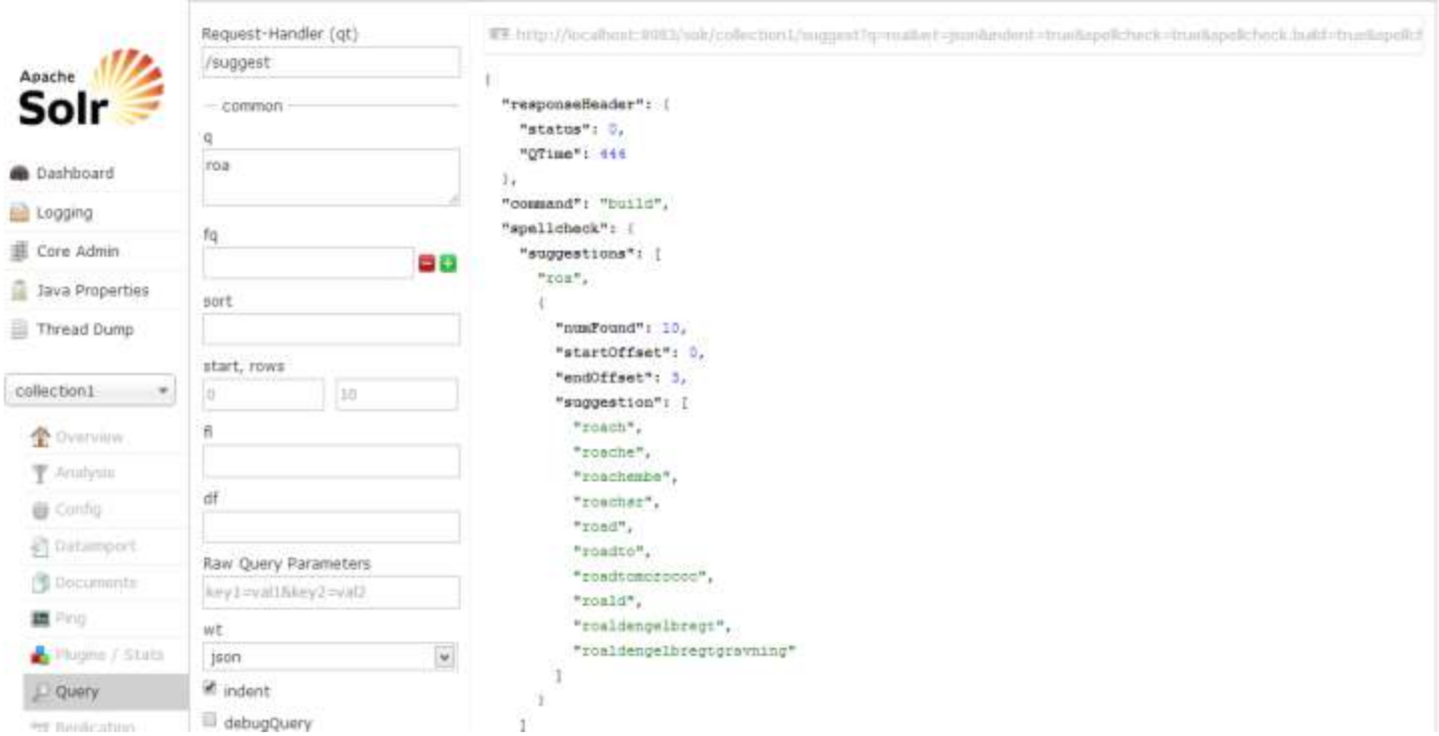
wt: json

☑ indent
□ debugQuery

□ dismax
□ edismax
□ hl
□ facet
□ spatial
☑ spellcheck
□ spellcheck.build
□ spellcheck.reload
spellcheck.q
spellcheck.dictionary
spellcheck.count
□ spellcheck.onlyMorePopular

```
  },
  "spellcheck": {
    "suggestions": [
      "amitab",
      {
        "numFound": 1,
        "startOffset": 0,
        "endOffset": 6,
        "suggestion": [
          "amitabh"
        ]
      },
      "bachchn",
      {
        "numFound": 1,
        "startOffset": 7,
        "endOffset": 14,
        "suggestion": [
          "bachchan"
        ]
      },
      "collation",
      "amitabh bachchan"
    ]
  }
}
```

- **Auto Suggest**
  When user will be in the middle of typing, the UI will suggest him/her the complete words which he/she might wish to write.



Request-Handler (qt): /suggest
q: roa
fq:
sort:
start, rows: 0 10
fl:
df:
Raw Query Parameters: key1=val1&key2=val2
wt: json
☑ indent
□ debugQuery

http://localhost:8983/solr/collection1/suggest?q=roa&wt=json&indent=true&spellcheck=true&spellcheck.build=true&spellc

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 444
  },
  "command": "build",
  "spellcheck": {
    "suggestions": [
      "roa",
      {
        "numFound": 10,
        "startOffset": 0,
        "endOffset": 3,
        "suggestion": [
          "roach",
          "roache",
          "roachembe",
          "roacher",
          "road",
          "roadto",
          "roadtomorrowcc",
          "roald",
          "roaldengelbregt",
          "roaldengelbregtgravning"
        ]
      }
    ]
```

- **More on this**
  The UI will suggest the user to know more about the subject which the user has searched for. This will display the other information present in the Infobox of the subject.

- **Name Search**
  If a user is not sure of what to search for about a subject, he/she can simply type the name and the entire information given in the Infobox will be displayed on UI.

CONFIGURATION DETAILS

Since Solr is a web-based search service, most operations take place by sending *HTTP GET* and *POST* requests from a client application to the Solr server. It provides client code for *Java, Ruby, Python,* and *PHP,* as well as the standard X*ML* responses that can easily be handled by any application. When Solr receives a request from the client, it parses the URL and routes the request to the appropriate *SolrRequestHandler*. The SolrRequestHandler is then responsible for processing the request parameters, doing any necessary computation, and assembling a *SolrQueryResponse*. After a response has been created, a *Query-ResponseWriter* implementation serializes the response and it's sent back to the client. Solr supports many different response formats, including *XML* and *JSON*, as well as formats that are easily used by languages like Ruby and PHP. Finally, custom *Query-ResponseWriters* can be plugged in to provide alternate responses when needed. With regard to content processing.

In Solr an index is built of one or more Documents. A Document consists of one or more Fields. A field consists of a *name, content,* and *metadata* telling Solr how to handle the content.

Our search is based on the uniqueKey - *name*. Any data will be fetched with reference to the name field.

<uniqueKey>name</uniqueKey>

The two main field types used are:

- **text_general**

This field type has a reasonable, generic, cross language defaults: it tokenizes with StandardTokenizer, removes stop words from case-sensitive "stpowords.txt".

Name has also been assigned text_general so that we can apply tokenizer rules to the field data.

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

- **string**

This field does not undergo tokenizer rules, hence fields like abbreviation, employer etc.

e.g. *<field name="education" type="string" indexed="true" stored="true" required="false" />*

Following filters have been applied to text_general field type:

- **solr.WhiteSpaceTokenizerFactory**

  This divides text at whitespaces, as defined by *java.lang.Character.isWhiteSpace()*. Adjacent sequences of non-whitespace characters form tokens.

  e.g. Alma mater is tokenized into 'Alma' and 'mater'.

- **solr.PatternReplaceFilterFactory**

  This takes two arguments: 1. The string which has to be replaced and 2. The pattern by which the string has to be replaced.

  e.g. ("@gmail.com", "***") will change abc@gmail.com to abc***.

- **solr.WordDelimiterFilterFactory**
  It splits words into subwords and performs optional transformations on subword groups. It also helps match words with different delimiters. One way of doing so is to specify *generateWordParts="1"* *catenateWords="1"* in the analyzer used for indexing, and *generateWordParts="1"* in the analyzer used for querying. By default, words are split into subwords with the following rules:

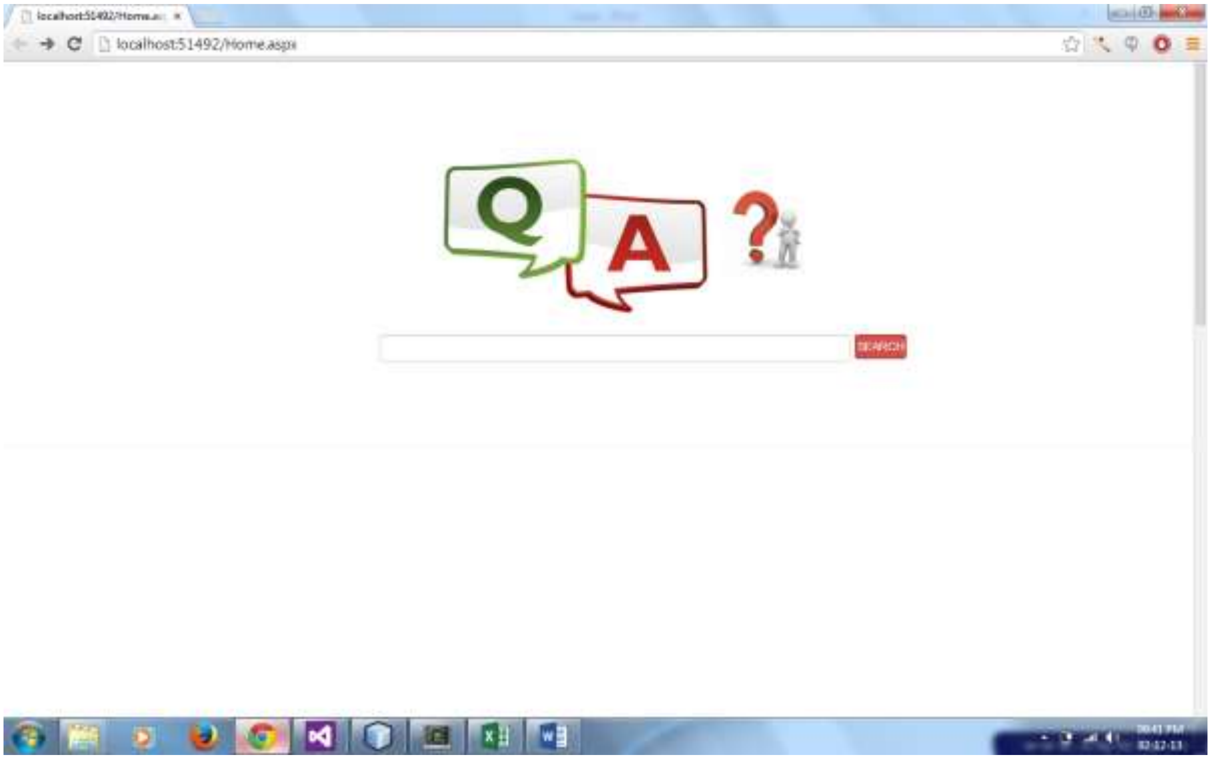
  - split on intra-word delimiters (all non alpha-numeric characters).
    "Wi-Fi" -> "Wi", "Fi"
  - split on case transitions (can be turned off)
    "PowerShot" -> "Power", "Shot"
  - split on letter-number transitions (can be turned off)
    "SD500" -> "SD", "500"
  - leading and trailing intra-word delimiters on each subword are ignored
    "//hello---there, 'dude'" -> "hello", "there", "dude"
  - trailing "'s" are removed for each subword (can be turned off)
    "O'Neil's" -> "O", "Neil"

- **solr.StopFilterFactory**
  This filter discards stop words as listed in stopwords.txt file. Stop words file should be in /solr/conf.
  <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt" enablePositionIncrements="true" />
  e.g. Where did Allan Dwan die? -> Where Allan Dwan die?

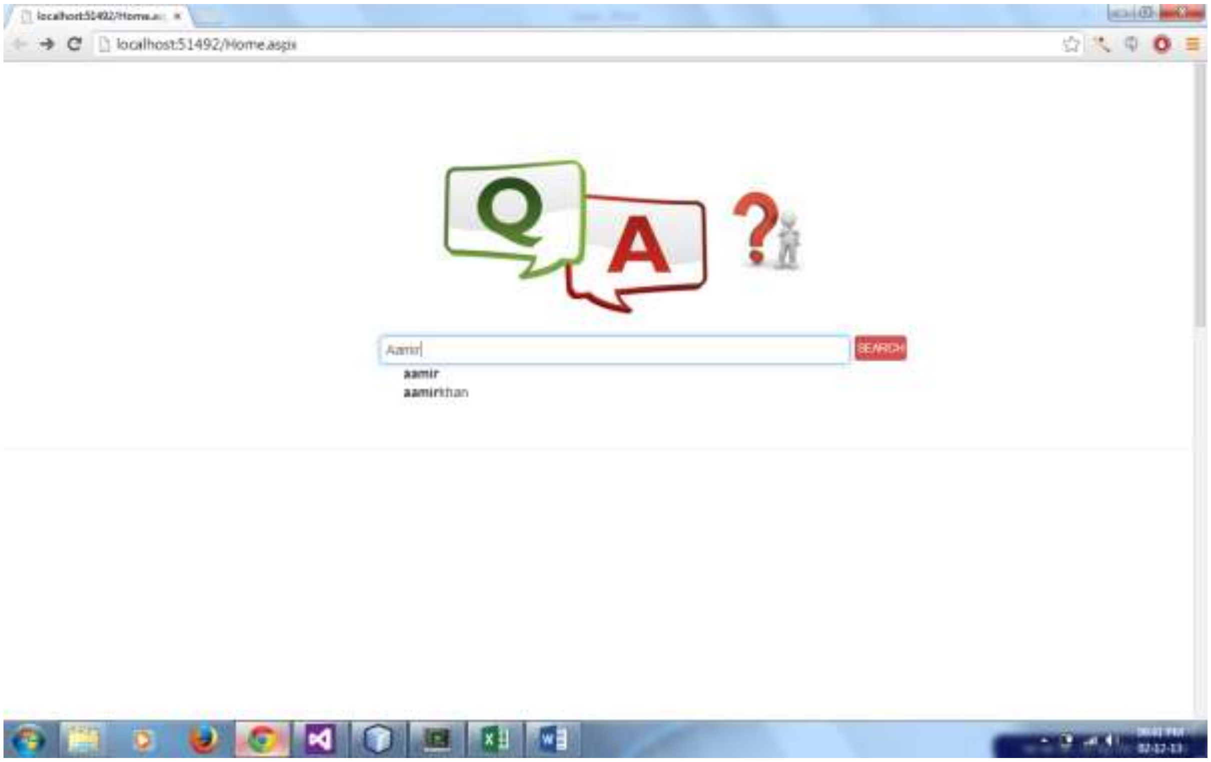
- **solr.LowerCaseFilterFactory**
  This filter changes all text to lower case.
  e.g. What is Alfred's Alma mater? -> what is alfred's alma mater?

UI SCREENSHOTS

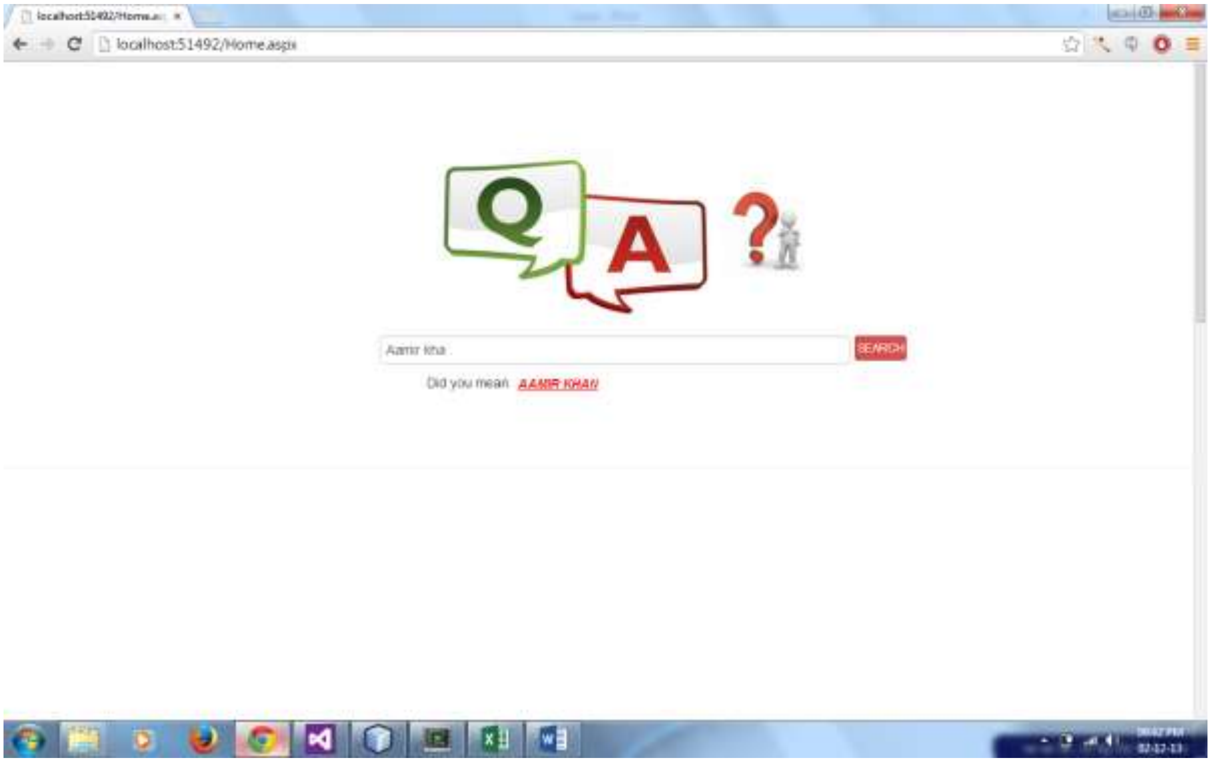
UI Screen looks like:



Auto Suggest:



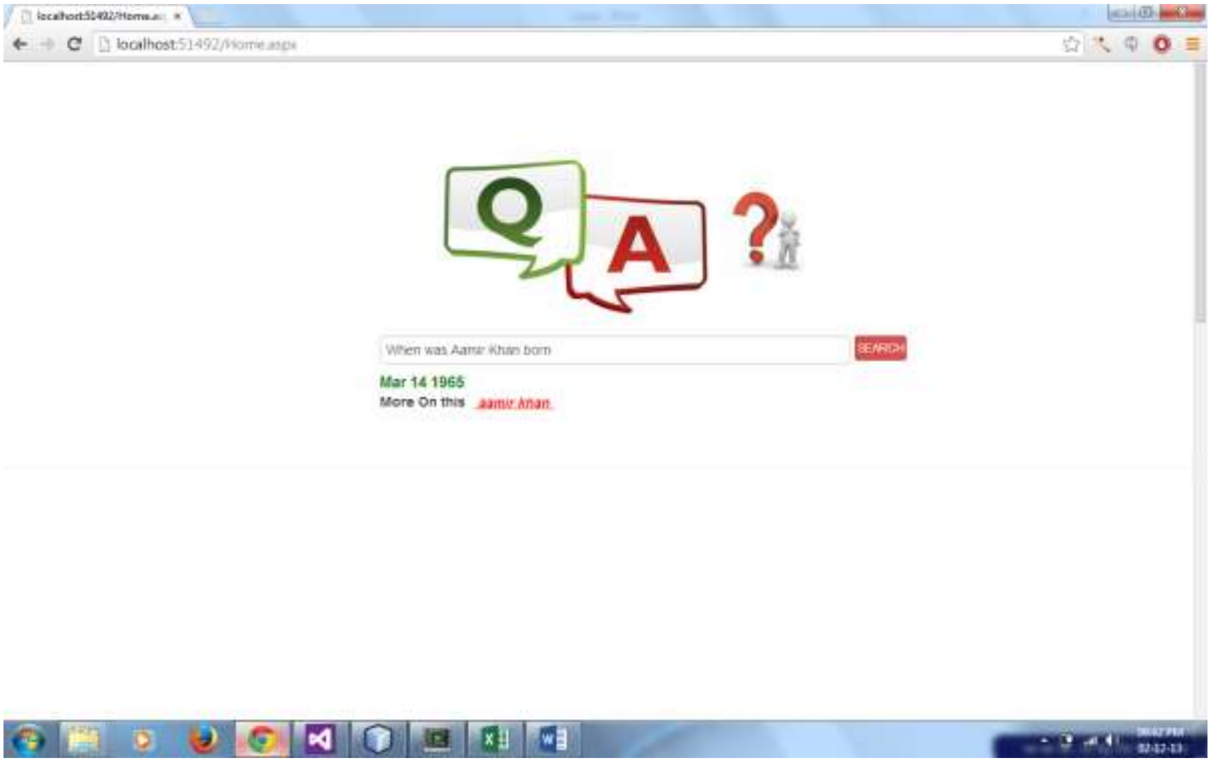
Spell Check:



Aamir kha .

Did you mean AAMIR KHAN

Response to the query, More on this:

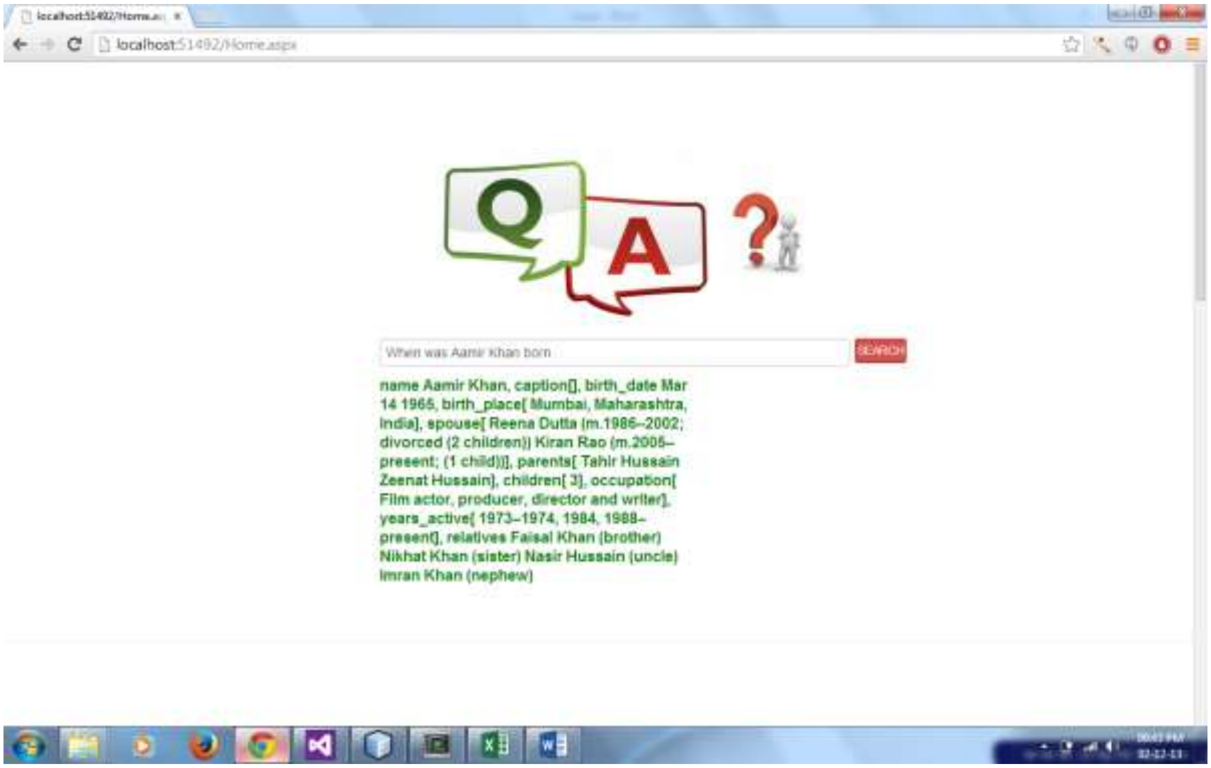


When was Aamir Khan born

Mar 14 1965

More On this aamir khan

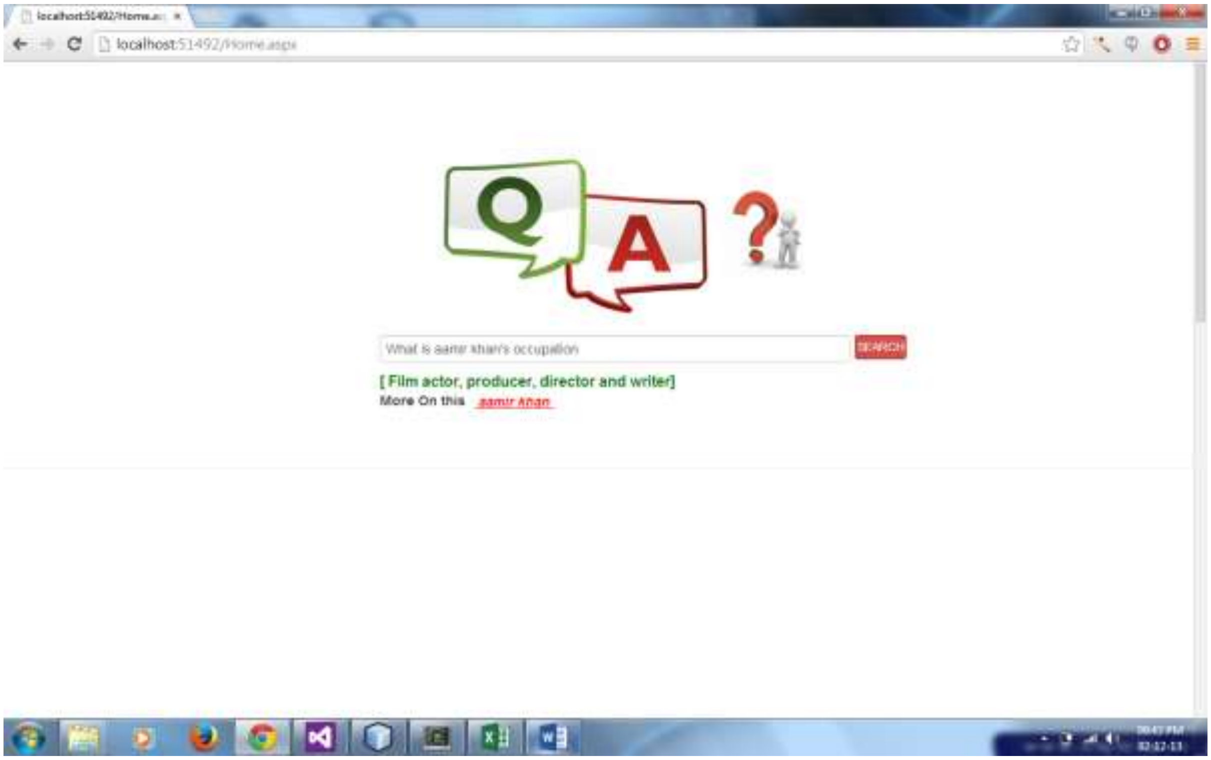
Response to ‘More on this’:



When was Aamir Khan born

name Aamir Khan, caption[], birth_date Mar 14 1965, birth_place[Mumbai, Maharashtra, India], spouse[Reena Dutta (m.1986–2002; divorced (2 children)) Kiran Rao (m.2005–present; (1 child))], parents[Tahir Hussain Zeenat Hussain], children[3], occupation[Film actor, producer, director and writer], years_active[1973–1974, 1984, 1988–present[], relatives Faisal Khan (brother) Nikhat Khan (sister) Nasir Hussain (uncle) Imran Khan (nephew)

Response to another query:

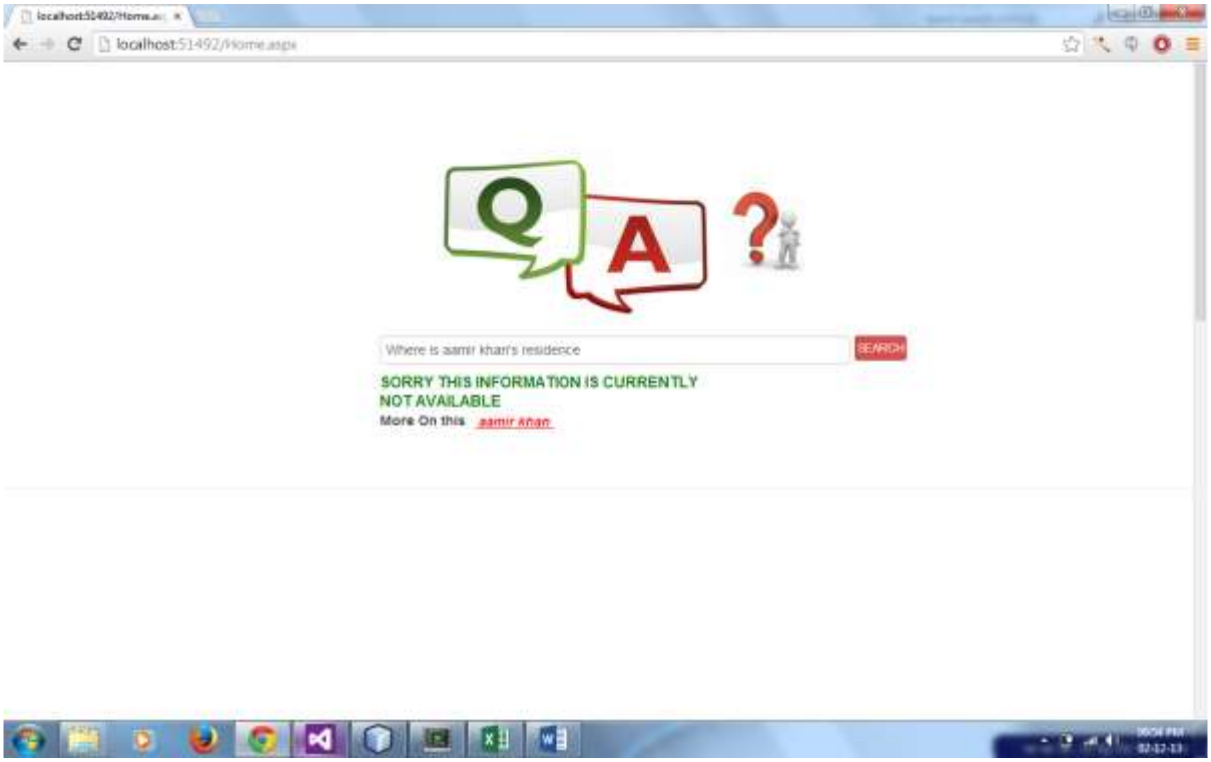


What is aamir khan's occupation

[Film actor, producer, director and writer]

More On this aamir khan

Response to a query for which information is not available:

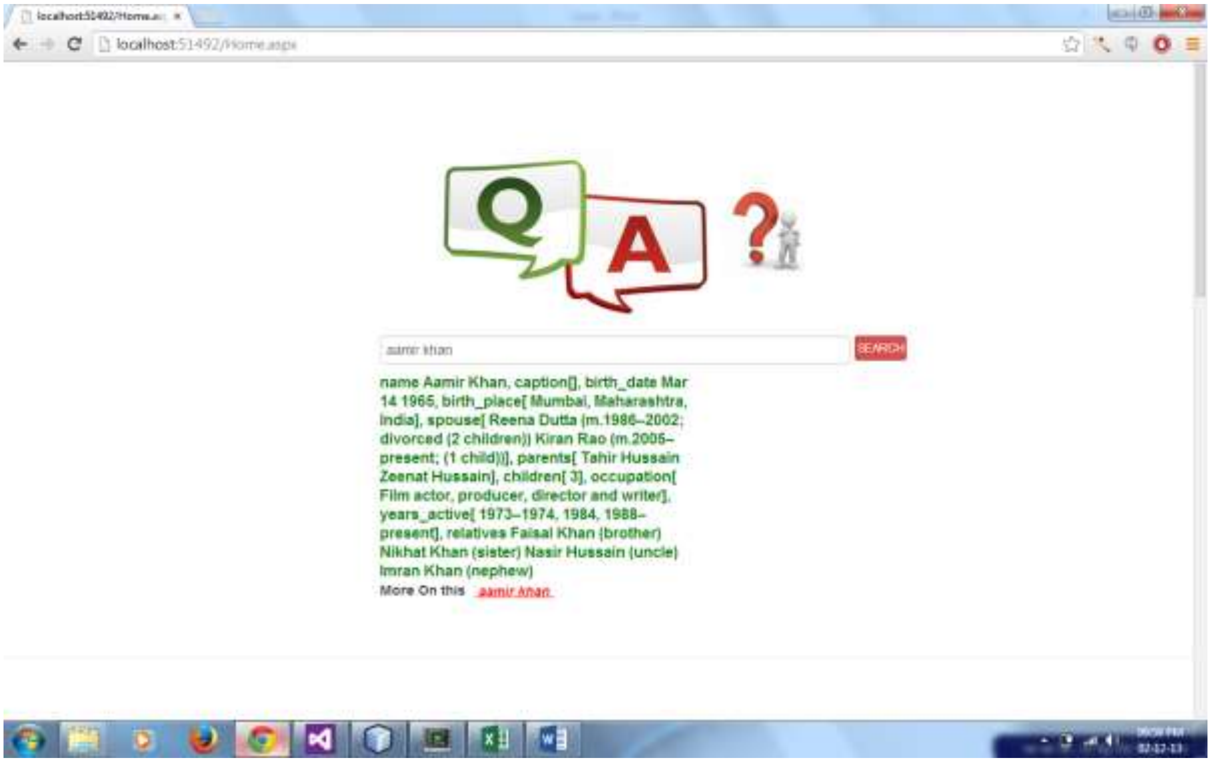


Where is aamir khan's residence

SORRY THIS INFORMATION IS CURRENTLY NOT AVAILABLE

More On this aamir khan

Name Search:



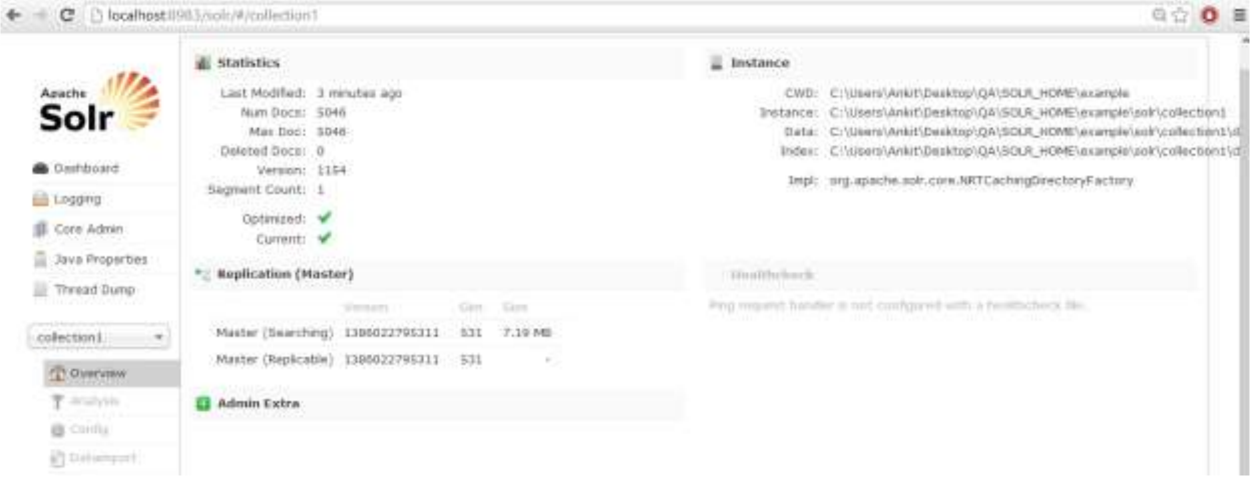
aamir khan

name Aamir Khan, caption[], birth_date Mar 14 1965, birth_place[Mumbai, Maharashtra, India], spouse[Reena Dutta (m.1986–2002; divorced (2 children)) Kiran Rao (m.2005–present; (1 child))], parents[Tahir Hussain Zeenat Hussain], children[3], occupation[Film actor, producer, director and writer], years_active[1973–1974, 1984, 1988–present], relatives Faisal Khan (brother) Nikhat Khan (sister) Nasir Hussain (uncle) Imran Khan (nephew)

More On this aamir khan

SOLR STATISTICS

Overview:



Statistics

Last Modified:	3 minutes ago
Num Docs:	5046
Max Doc:	5046
Deleted Docs:	0
Version:	1164
Segment Count:	1
Optimized:	✓
Current:	✓

Replication (Master)

	Version	Gen	Size
Master (Searching)	1386022795311	531	7.19 MB
Master (Replicable)	1386022795311	531	-

Admin Extra

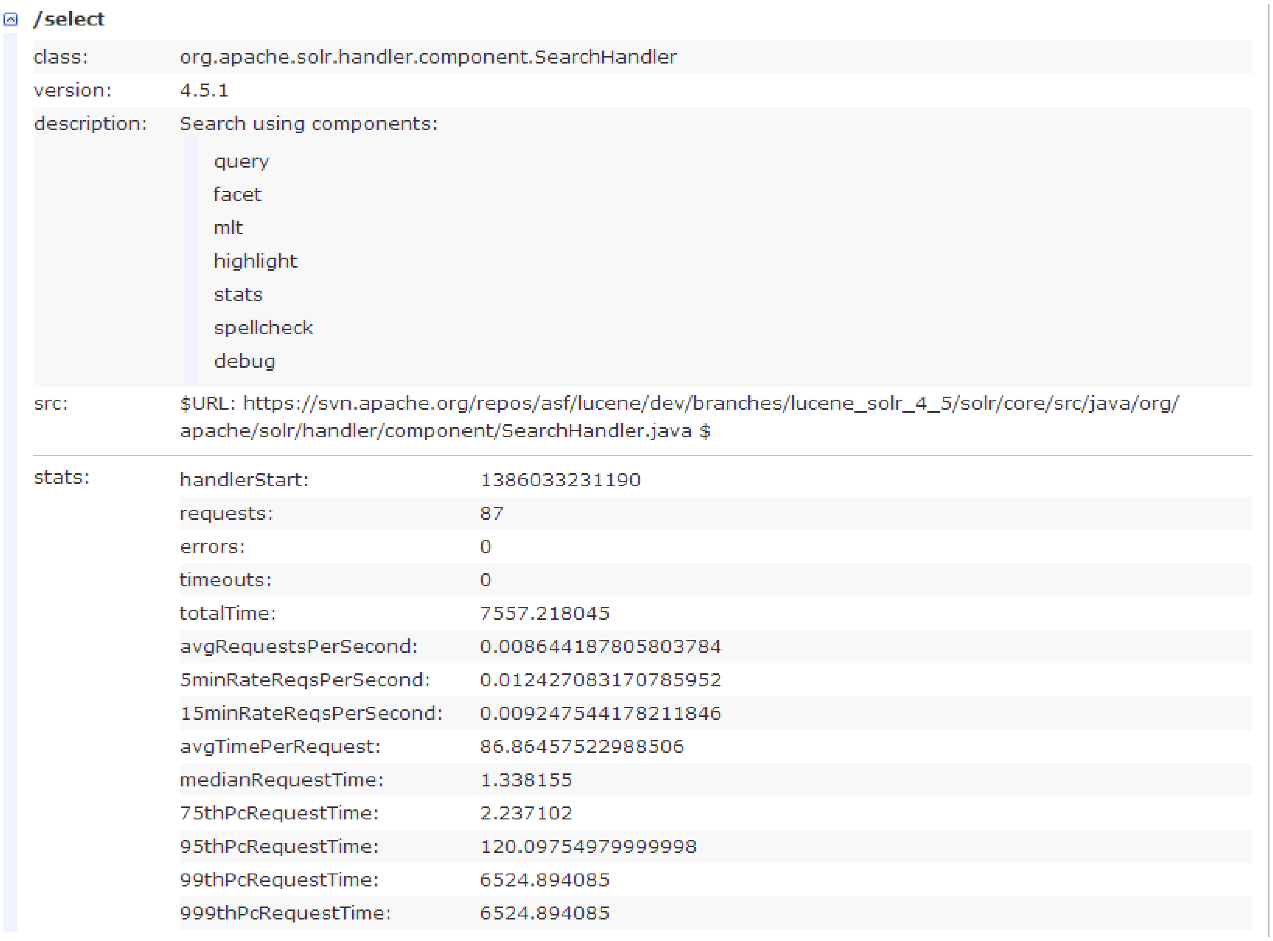
Instance

CWD:	C:\Users\Ankit\Desktop\QA\SOLR_HOME\example
Instance:	C:\Users\Ankit\Desktop\QA\SOLR_HOME\example\solr\collection1
Data:	C:\Users\Ankit\Desktop\QA\SOLR_HOME\example\solr\collection1\d
Index:	C:\Users\Ankit\Desktop\QA\SOLR_HOME\example\solr\collection1\d
Impl:	org.apache.solr.core.NRTCachingDirectoryFactory

Healthcheck

Ping request handler is not configured with a healthcheck file.

Select Query: (Average Response Time)



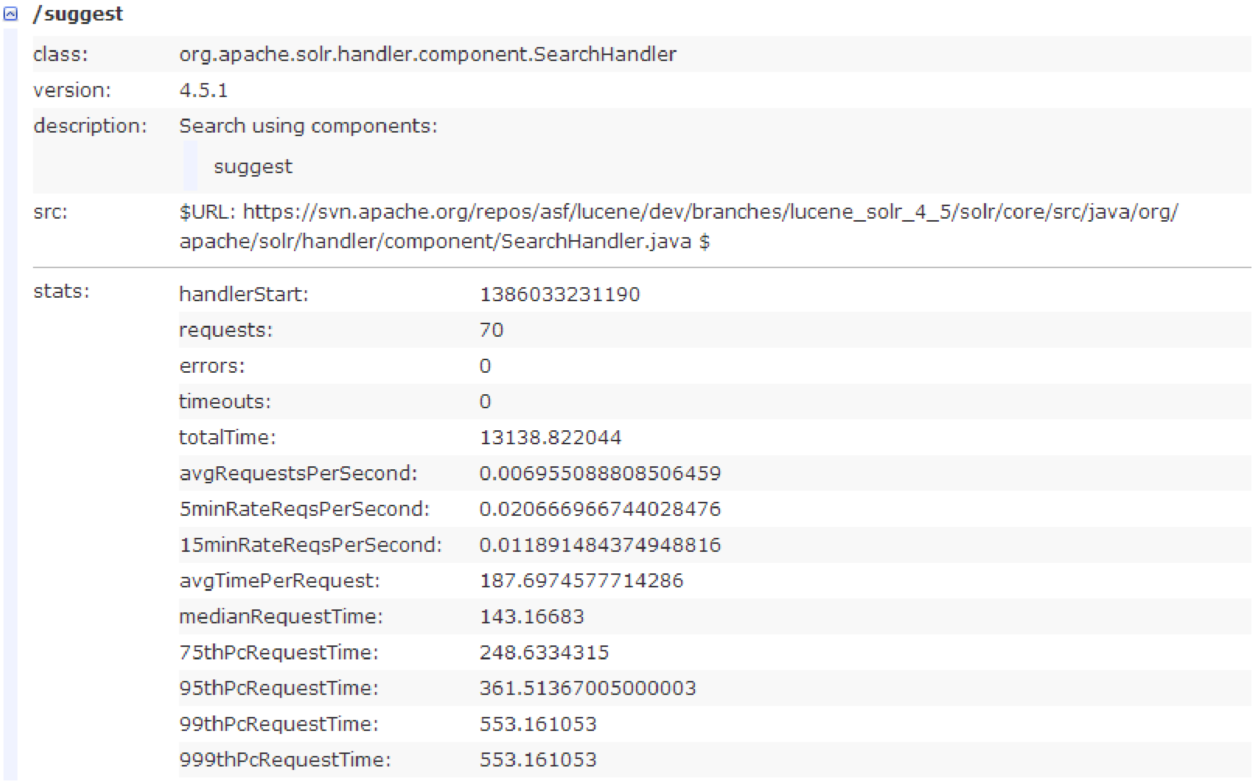
/select

class:	org.apache.solr.handler.component.SearchHandler
version:	4.5.1
description:	Search using components: query facet mlt highlight stats spellcheck debug
src:	$URL: https://svn.apache.org/repos/asf/lucene/dev/branches/lucene_solr_4_5/solr/core/src/java/org/apache/solr/handler/component/SearchHandler.java $

stats:

handlerStart:	1386033231190
requests:	87
errors:	0
timeouts:	0
totalTime:	7557.218045
avgRequestsPerSecond:	0.008644187805803784
5minRateReqsPerSecond:	0.012427083170785952
15minRateReqsPerSecond:	0.009247544178211846
avgTimePerRequest:	86.86457522988506
medianRequestTime:	1.338155
75thPcRequestTime:	2.237102
95thPcRequestTime:	120.09754979999998
99thPcRequestTime:	6524.894085
999thPcRequestTime:	6524.894085

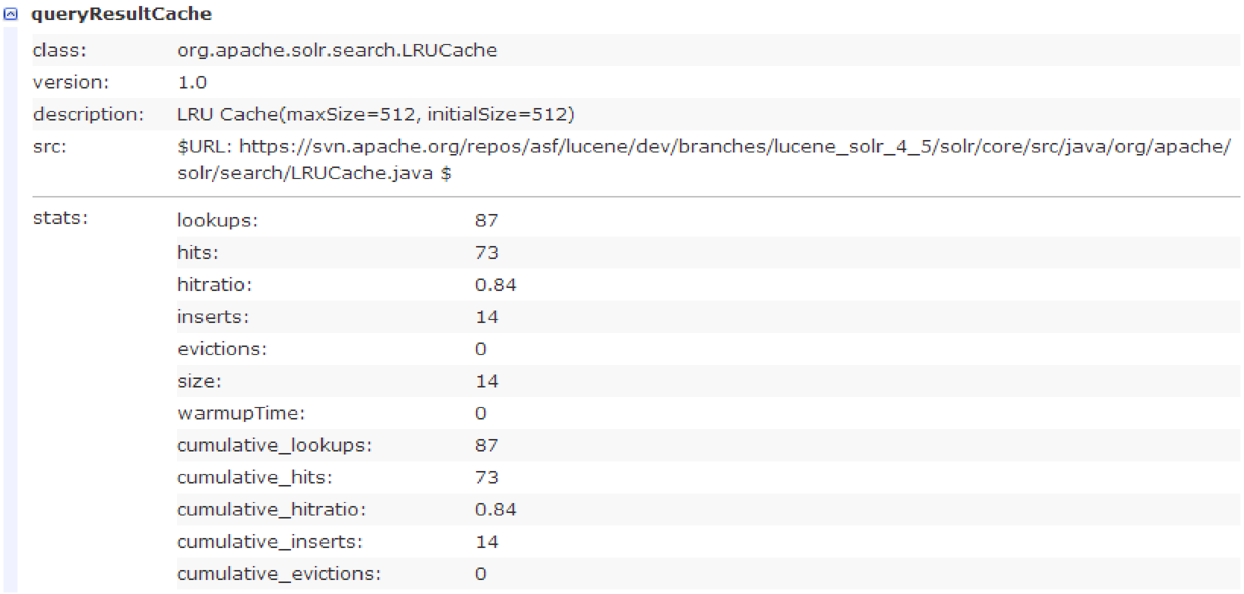
Suggest Query (Average Response Time)



/suggest

class:	org.apache.solr.handler.component.SearchHandler	
version:	4.5.1	
description:	Search using components: suggest	
src:	$URL: https://svn.apache.org/repos/asf/lucene/dev/branches/lucene_solr_4_5/solr/core/src/java/org/apache/solr/handler/component/SearchHandler.java $	
stats:	handlerStart:	1386033231190
	requests:	70
	errors:	0
	timeouts:	0
	totalTime:	13138.822044
	avgRequestsPerSecond:	0.006955088808506459
	5minRateReqsPerSecond:	0.020666966744028476
	15minRateReqsPerSecond:	0.011891484374948816
	avgTimePerRequest:	187.6974577714286
	medianRequestTime:	143.16683
	75thPcRequestTime:	248.6334315
	95thPcRequestTime:	361.51367005000003
	99thPcRequestTime:	553.161053
	999thPcRequestTime:	553.161053

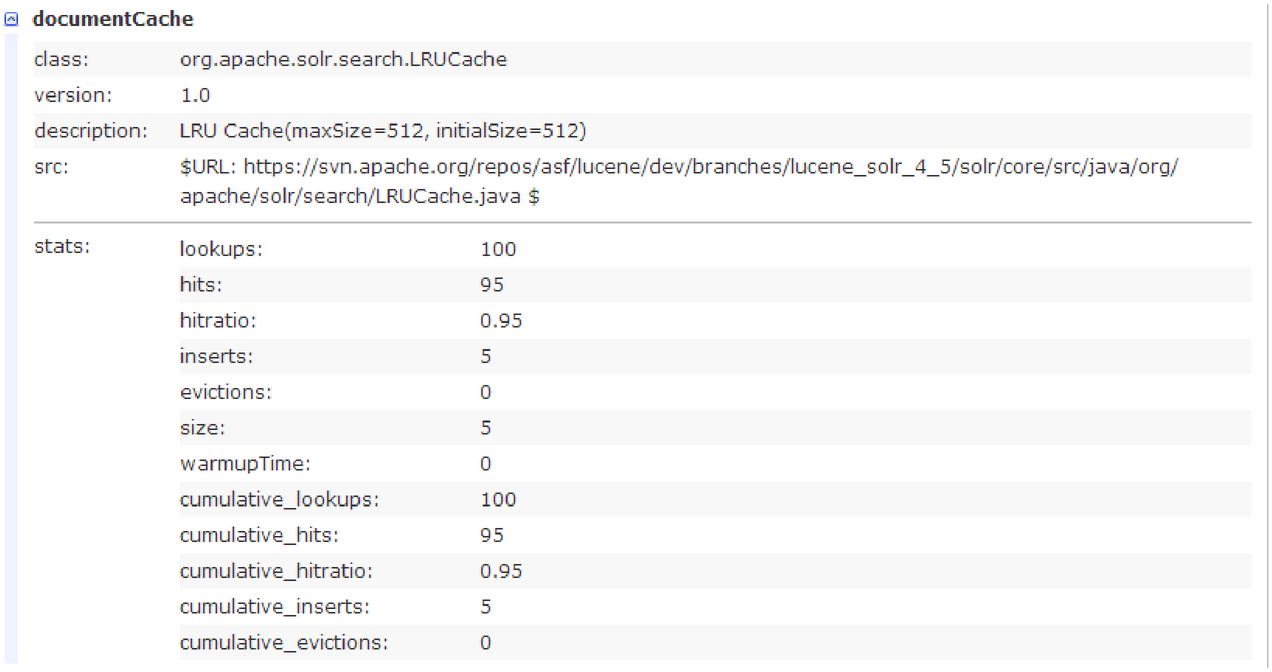
Hits (Query Hits)



queryResultCache

class:	org.apache.solr.search.LRUCache	
version:	1.0	
description:	LRU Cache(maxSize=512, initialSize=512)	
src:	$URL: https://svn.apache.org/repos/asf/lucene/dev/branches/lucene_solr_4_5/solr/core/src/java/org/apache/solr/search/LRUCache.java $	
stats:	lookups:	87
	hits:	73
	hitratio:	0.84
	inserts:	14
	evictions:	0
	size:	14
	warmupTime:	0
	cumulative_lookups:	87
	cumulative_hits:	73
	cumulative_hitratio:	0.84
	cumulative_inserts:	14
	cumulative_evictions:	0

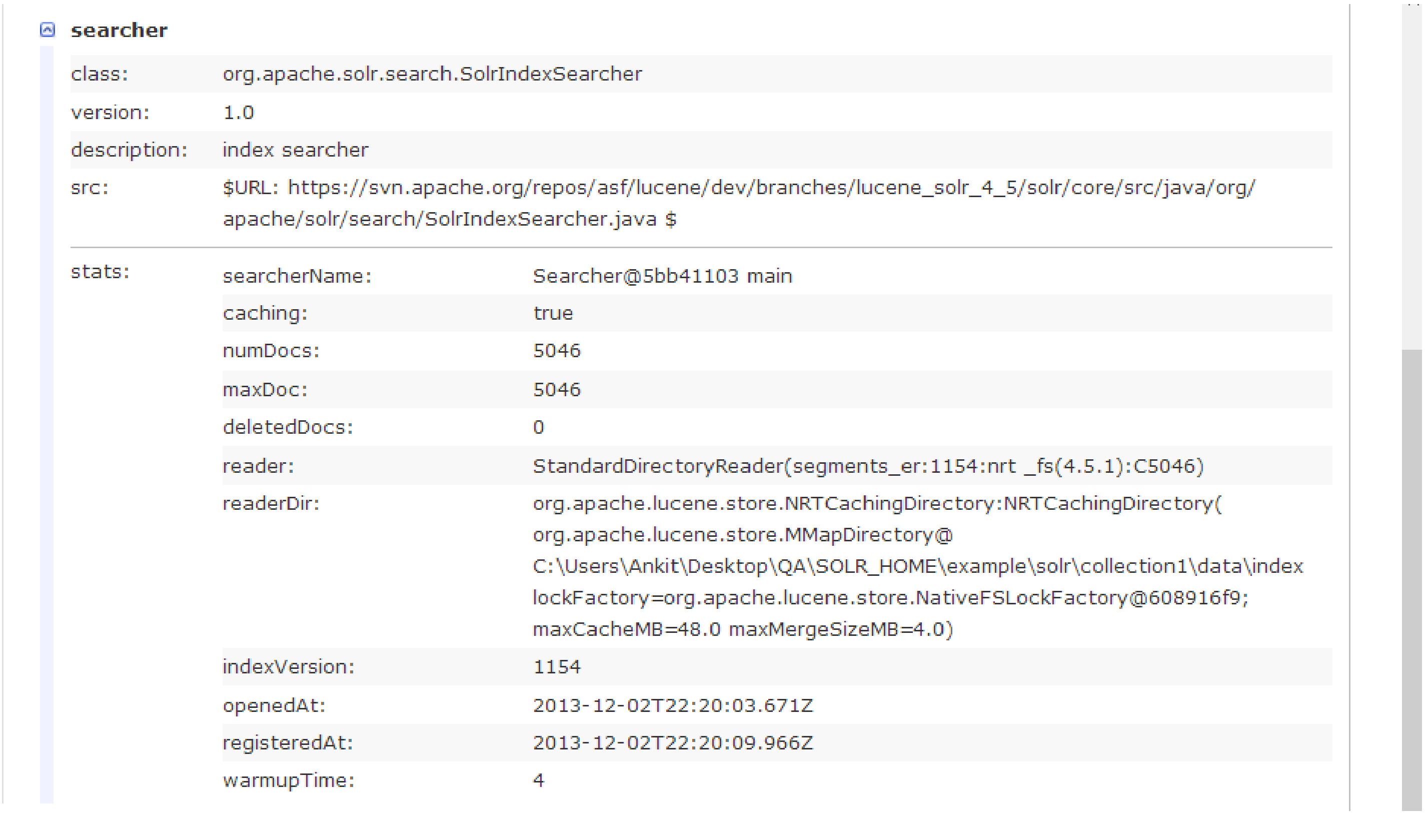
Hits (Document Hits):



documentCache

class:	org.apache.solr.search.LRUCache	
version:	1.0	
description:	LRU Cache(maxSize=512, initialSize=512)	
src:	$URL: https://svn.apache.org/repos/asf/lucene/dev/branches/lucene_solr_4_5/solr/core/src/java/org/apache/solr/search/LRUCache.java $	
stats:	lookups:	100
	hits:	95
	hitratio:	0.95
	inserts:	5
	evictions:	0
	size:	5
	warmupTime:	0
	cumulative_lookups:	100
	cumulative_hits:	95
	cumulative_hitratio:	0.95
	cumulative_inserts:	5
	cumulative_evictions:	0

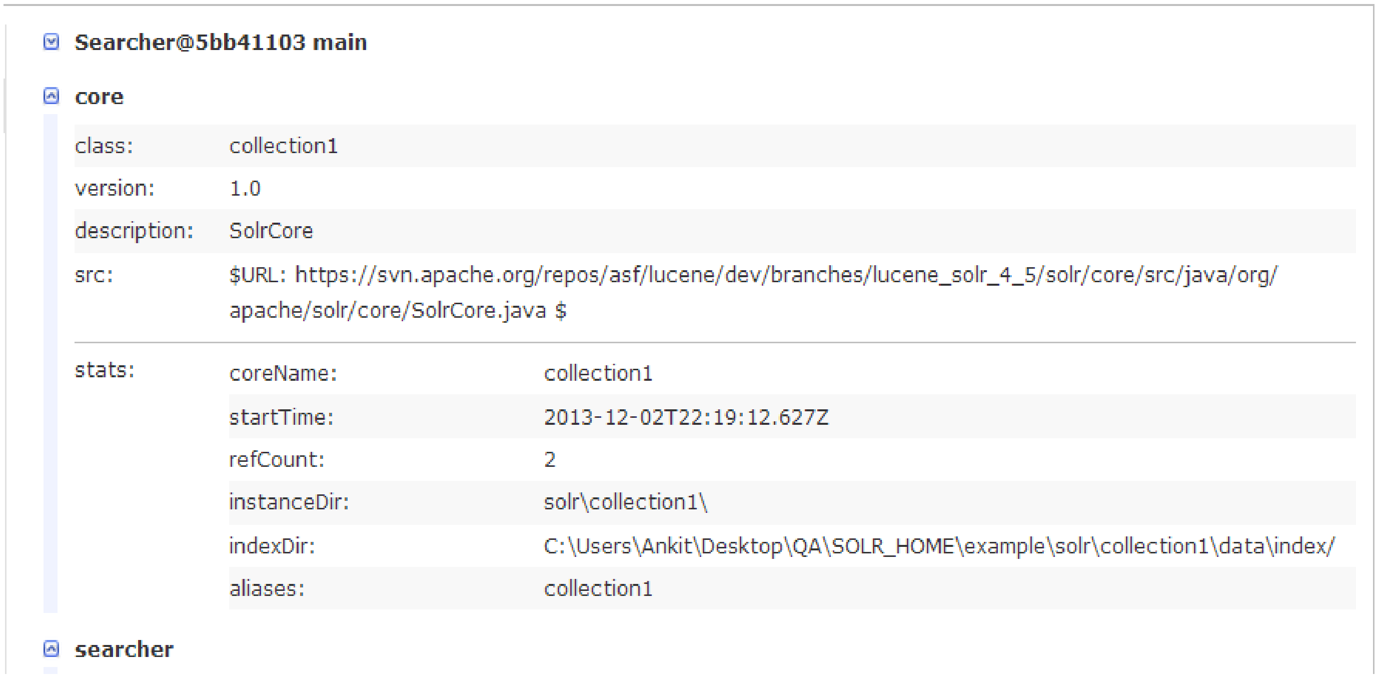
Warm up Time:



searcher

class:	org.apache.solr.search.SolrIndexSearcher	
version:	1.0	
description:	index searcher	
src:	$URL: https://svn.apache.org/repos/asf/lucene/dev/branches/lucene_solr_4_5/solr/core/src/java/org/apache/solr/search/SolrIndexSearcher.java $	
stats:	searcherName:	Searcher@5bb41103 main
	caching:	true
	numDocs:	5046
	maxDoc:	5046
	deletedDocs:	0
	reader:	StandardDirectoryReader(segments_er:1154:nrt _fs(4.5.1):C5046)
	readerDir:	org.apache.lucene.store.NRTCachingDirectory:NRTCachingDirectory(org.apache.lucene.store.MMapDirectory@C:\Users\Ankit\Desktop\QA\SOLR_HOME\example\solr\collection1\data\index lockFactory=org.apache.lucene.store.NativeFSLockFactory@608916f9; maxCacheMB=48.0 maxMergeSizeMB=4.0)
	indexVersion:	1154
	openedAt:	2013-12-02T22:20:03.671Z
	registeredAt:	2013-12-02T22:20:09.966Z
	warmupTime:	4

Core:



Searcher@5bb41103 main

core

class:	collection1	
version:	1.0	
description:	SolrCore	
src:	$URL: https://svn.apache.org/repos/asf/lucene/dev/branches/lucene_solr_4_5/solr/core/src/java/org/apache/solr/core/SolrCore.java $	
stats:	coreName:	collection1
	startTime:	2013-12-02T22:19:12.627Z
	refCount:	2
	instanceDir:	solr\collection1\
	indexDir:	C:\Users\Ankit\Desktop\QA\SOLR_HOME\example\solr\collection1\data\index/
	aliases:	collection1

searcher

FUTURE WORK

- In future we intend to expand the question set to "why" and "how" which will involve more complex natural processing techniques.

- In the current implementation, models used are restricted to identifying person, organization and film. More models can also be trained and incorporated into the application which can work for restricted domains as well as identifying other entities. This will boost the accuracy further and also increase the efficiency.

- Presently, our system accepts query in the format of a question, in future we wish to make the query completely free text.

MEMBER CONTRIBUTION

Infobox Parsing, Query Processing	Pranav Gupta
Schema.xml, solrConfig.xml, SolrJ (Indexing)	Ankit Jain, Milky Sahu
User Interface, Additional Features	Deepak Veerupapuram