COMP 631: Introduction to Information Retrieval

XIA (BEN) HU

CS, Rice University

https://cs.rice.edu/~xh37/index.html

Point distribution

- Class participation and quizzes - 5%
- Three homework assignments -20%
- Project - 30%
- Three Exams - 45%

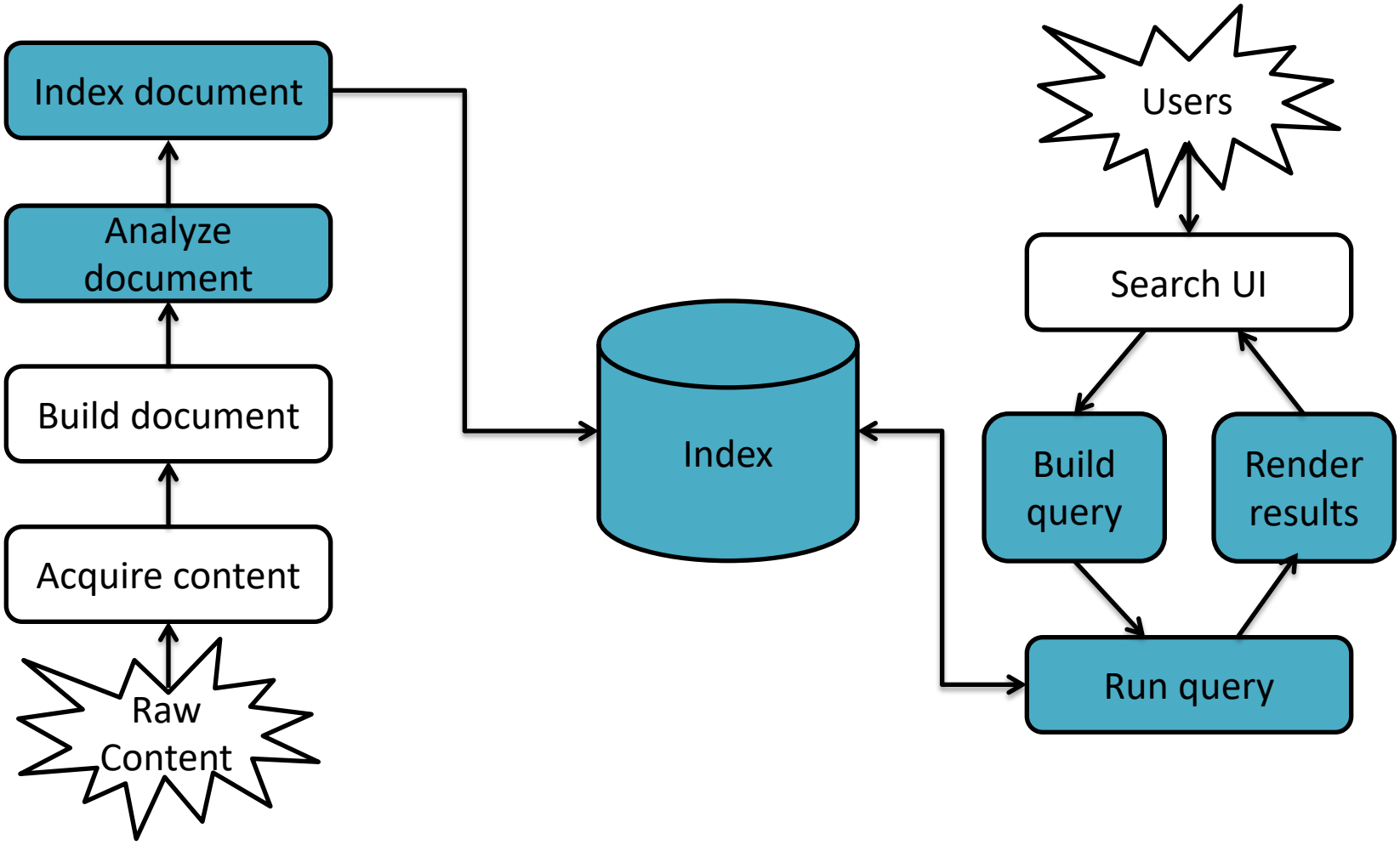
- Late penalty, **YES**, increasing *exponentially* wrt the number of days. Late = Original / 2^n , $n > 0$.
- Academic integrity

On my honor, I have neither given nor received any unauthorized aid on this (exam, quiz, paper)

Project

- Team Project
  - Real-world application
  - Three checkpoints, including data crawling, search engine, and application
  - Progress report
  - Final report
  - Class presentation and/or demo

CSCE 670 Project – An IR System



Format

- Group Project (preferably 3 students)
  - Task 1: Data crawling (20%)
  - Task 2: Building a search engine (20%)
  - Task 3: An application (20%)
  - Final report (20%)
  - Video demo (20%)

Task 1: Data Crawling

Task 1: Data Crawling

- **Select an open site with text information**
  - Post site's name (e.g., Wikipedia) on Canvas (if you are the first group) under the Discussion → Project Selection
  - Select the site by replying to the site's name, with names of group members
  - Only 5 groups can crawl the same site (first come, first serve)
  - Please do not post anything unrelated to site selection and group members under the discussion "Project Selection"
  - If you decide to change your site report it again and remove your reply from the old post; otherwise, your first submission is considered as your site
- API or Parsing. Many sites have API limits if too many requests are sent to their servers.

Propose a problem you would like to solve

- Example problems:
  - Sentiment analysis for Amazon: Given a movie/product/people, predict whether people are happy or not
  - Recommendation in NFTs: Given a particular NFT (non-fungible tokens) description, recommend related instances (i.e. Open Sea)
  - Visualization: Given a particular GPS location project Instagram photos/reels in a map that we can interact with

Propose a problem you would like to solve

- Something beyond (Not required, and Solr cannot support these applications):
  - Recommendation in sustainability: given a photo taken of an object (ideally trash) recommend videos and websites for reuse or recycling
  - Visualization: Given a particular GPS location project Instagram photos/reels in a map that we can interact with
  - Recommendation: given a video/picture automatically annotate it with popular (most-liked) comments from similar TikToks/Instagram posts.

How Can I find a Problem?

- SIGIR Demo Papers
  - http://sigir.org/sigir2014/finaldemos.php
  - http://www.sigir.org/sigir2013/demonstrations.html
- Other conferences
  - CIKM, ECIR

Submit a Proposal

- Write 2 pages. When writing the proposal you should try to answer the following questions:
  - What is the problem you are solving?
  - What data will you use?
  - What work do you plan to do the project?
  - Which algorithms/techniques/models you plan to use/develop? Be as specific as you can!
  - Who will evaluate your method? How will you test it? How will you measure success?
  - What do you expect to submit/accomplish by the end of the semester?

Checklist for Task 1

- Report your site and your group members on Canvas
- A project proposal (for task 3) with no more than two pages on Canvas – Proposal means preliminary
  - UIN-UIN-UIN-proposal.pdf
- Report of Task 1: no more than two pages on Canvas
  - UIN-UIN-UIN-task1.pdf

Submissions for Task 1

- Submit a zip file containing all of the documents you crawled to Canvas
  - Zip file name: UIN-UIN-UIN.zip
  - At least 100k documents (not hard requirement)
  - If it is full documents, such as Wikipedia, each file only contains one paragraph from the original document. If it is tweet or post (<140 chars), submit the full docs.

- The deadline for this is **Feb 21st, 11:59pm**
- **Class on Feb 20th is saved for project**

Task 2: Search Engine

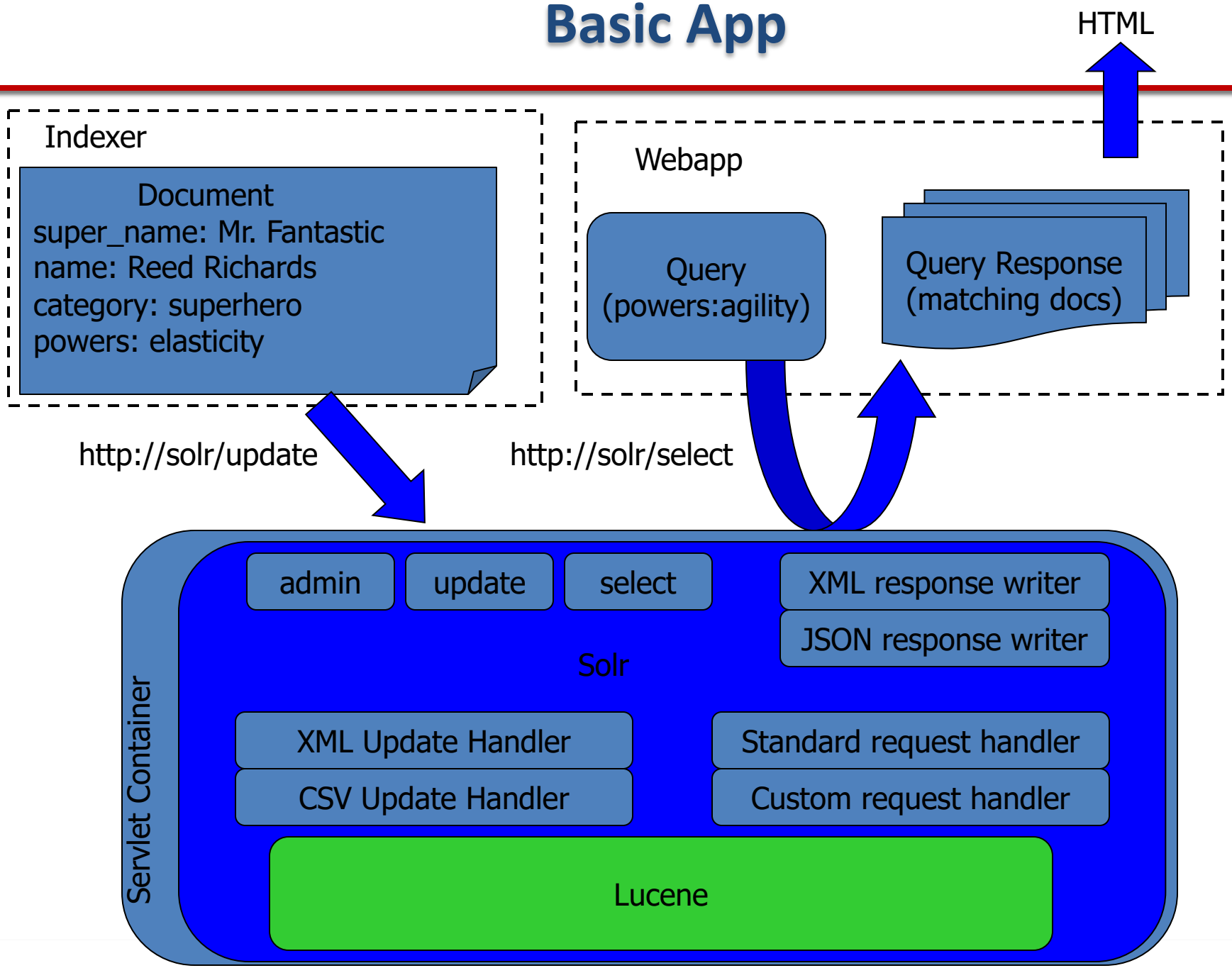
What is Lucene

- High performance, scalable, full-text search **library**
- Focus: Indexing + Searching Documents
  - "Document" is just a list of name+value pairs
- No crawlers or document parsing
- Flexible Text Analysis (tokenizers + token filters)
- 100% Java, no dependencies, no config files

What is Solr

- A full text search server based on Lucene
- XML/HTTP, JSON Interfaces
- Faceted Search (category counting)
- Flexible data schema to define types and fields
- Hit Highlighting
- Configurable Advanced Caching
- Index Replication
- Extensible Open Architecture, Plugins
- Web Administration Interface
- Written in Java5, deployable as a WAR

Basic App


HTML
Indexer
Document
super_name: Mr. Fantastic
name: Reed Richards
category: superhero
powers: elasticity
Webapp
Query
(powers:agility)
Query Response
(matching docs)
http://solr/update
http://solr/select
Servlet Container
admin
update
select
XML response writer
JSON response writer
Solr
XML Update Handler
CSV Update Handler
Standard request handler
Custom request handler
Lucene

Indexing Data

HTTP POST to http://localhost:8983/solr/update

```
<add><doc>
  <field name="id">05991</field>
  <field name="name">Peter Parker</field>
  <field name="supername">Spider-Man</field>
  <field name="category">superhero</field>
  <field name="powers">agility</field>
  <field name="powers">spider-sense</field>
</doc></add>
```

Data upload methods

```
URL=http://localhost:8983/solr/update/csv
```

- HTTP POST body (curl, HttpClient, etc)

```
curl $URL -H 'Content-type:text/plain;
  charset=utf-8' --data-binary @info.csv
```

- Multi-part file upload (browsers)
- Request parameter

```
?stream.body='Cyclops, Scott Summers,…'
```

- Streaming from URL (must enable)

```
?stream.url=file://data/info.csv
```

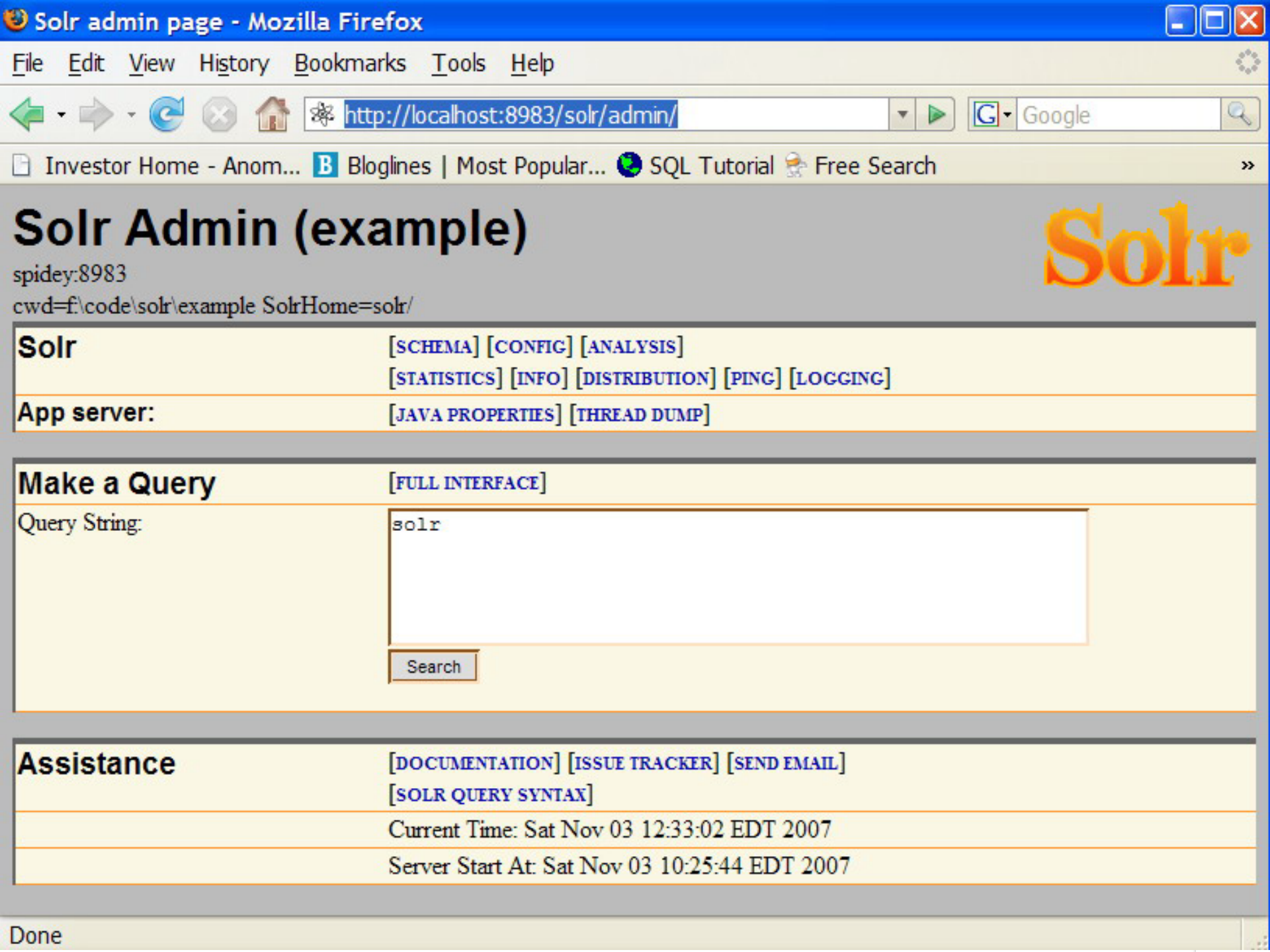
Indexing with SolrJ

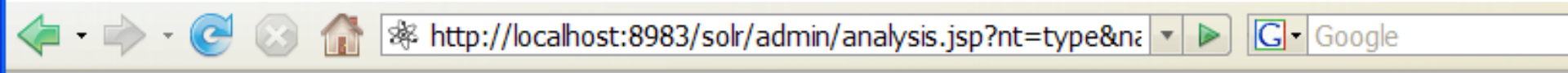
```
// Solr's Java Client API… remote or embedded/local!
SolrServer server = new
    CommonsHttpSolrServer("http://localhost:8983/solr
    ");

SolrInputDocument doc = new SolrInputDocument();
doc.addField("supername","Daredevil");
doc.addField("name","Matt Murdock");
doc.addField("category","superhero");

server.add(doc);
server.commit();
```

Searching

http://localhost:8983/solr/select?q=powers:agility&start=0&rows=2&fl=supername,category

```
<response>
 <result numFound="427" start="0">
  <doc>
   <str name="supername">Spider-Man</str>
   <str name="category">superhero</str>
  </doc>
  <doc>
   <str name="supername">Msytique</str>
   <str name="category">supervillain</str>
  </doc>
 </result>
</response>
```


Solr admin page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8983/solr/admin/
Google
Investor Home - Anom... Bloglines | Most Popular... SQL Tutorial Free Search
Solr Admin (example)
spidey:8983
cwd=f:\code\solr\example SolrHome=solr/
Solr
Solr [SCHEMA] [CONFIG] [ANALYSIS]
[STATISTICS] [INFO] [DISTRIBUTION] [PING] [LOGGING]
App server: [JAVA PROPERTIES] [THREAD DUMP]
Make a Query [FULL INTERFACE]
Query String: solr
Search
Assistance [DOCUMENTATION] [ISSUE TRACKER] [SEND EMAIL]
[SOLR QUERY SYNTAX]
Current Time: Sat Nov 03 12:33:02 EDT 2007
Server Start At: Sat Nov 03 10:25:44 EDT 2007
Done



Solr Admin (example)

spidey:8983

cwd=f:\code\solr\example SolrHome=solr/

Field Analysis

Field type: simple

Field value (Index)

verbose output ☐

highlight matches ☑

The Justice League saved the day by defeating Darkseid

Field value (Query)

verbose output ☐

the defeat of DarkSeid

Analyze

Index Analyzer

The	Justice	League	saved	the	day	by	defeating	Darkseid
Justice	League	saved	day	defeating	Darkseid			
justice	league	saved	day	defeating	darkseid			
justic	leagu	save	day	defeat	darkseid			

Query Analyzer

the	defeat	of	DarkSeid
defeat	DarkSeid		
defeat	darkseid		
defeat	darkseid		

Checklist for Task 2

- Report of Task 2: no more than two pages on Canvas
  - UIN-UIN-UIN-task2.pdf

- Video recording to show your interaction with the search engine
  - A link in a text file (web link or youtube link) on Canvas
  - Audio is not required
  - Show your IDs in the video or captions
  - No more than five minutes
  - UIN-UIN-UIN-solr.txt

- The deadline for this is **March 23rd, 11:59pm**

Task 3: Application

Checklist for Task 3

- Final report: no more than six pages on Canvas
  - UIN-UIN-UIN-project.pdf
  - Reports for task1 and task2 can be re-used

- Video recording to show your demo
  - A link in a text file (web link or youtube link) on Canvas
  - Audio is required
  - Show your IDs in the video or captions
  - No more than five minutes
  - UIN-UIN-UIN-demo.txt

  - The deadline for this is **April 21st, 11:59pm**

Project Report

- **Introduction/Motivation/Problem Definition (25%)**
  What is it that you are trying to solve/achieve and why does it matter.

- **Prior Work (10%)**
  How does your project relate to previous work. Please give a short summary on each paper you cite and include how it is relevant.

- **Model/Algorithm/Method (20%)**
  This is where you give a detailed description of your primary contribution. It is especially important that this part be clear and well written so that we can fully understand what you did.

- **Results and findings (25%)**

- **Style and writing (20%)**
  Overall writing, grammar, organization and neatness.