Solr Primer
Day 1

Macy’s

jeffnb@gmail.com

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Links

- Download Solr:
  - http://bit.ly/1WGzhYy
- Github Repo
  - https://github.com/jeffnb/solr-lab-intro

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Course Outline [Day 1]

- Introduction to Solr
- Solr Overview
- Solr Admin
- Searching
- SolrJ Searching

- Project: Search and Faceting

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Course Outline [Day 2]

- Schema
- Indexing
- Configuration

- Lab

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Course Outline [Day 3]

- Configuration continued
- Optimization
- SolrCloud
- NLP

- Lab

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Part 0: Solr

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: The Database Issue

- Filtering simple things

```
SELECT * FROM movies WHERE Year = 1995;
```

- A little more complicated looking in multiple fields

```
SELECT * FROM movies WHERE Title like ‘%Toy%’
   OR Plot like ‘%Toy%’;
```

- Even more so if we need to check many fields

```
SELECT * FROM movies WHERE Title like ‘%Tom%’
   OR Plot like ‘%Tom%’ OR Actors like ‘%Tom%’
   OR Director like ‘%Tom%’;
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: The Database Issue

- Text field searching is slow
- No “best” matches
- “Try” != “Tries”

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr

- What?
  - Enterprise Level Search Engine
- Why?
  - Extremely fast
  - Extremely flexible
  - Extremely powerful
  - Extremely scalable

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: What Can it Do

- Searching
- Faceting
- Tokenizing
- Spell Checking
- Replicating
- Similarity-ing

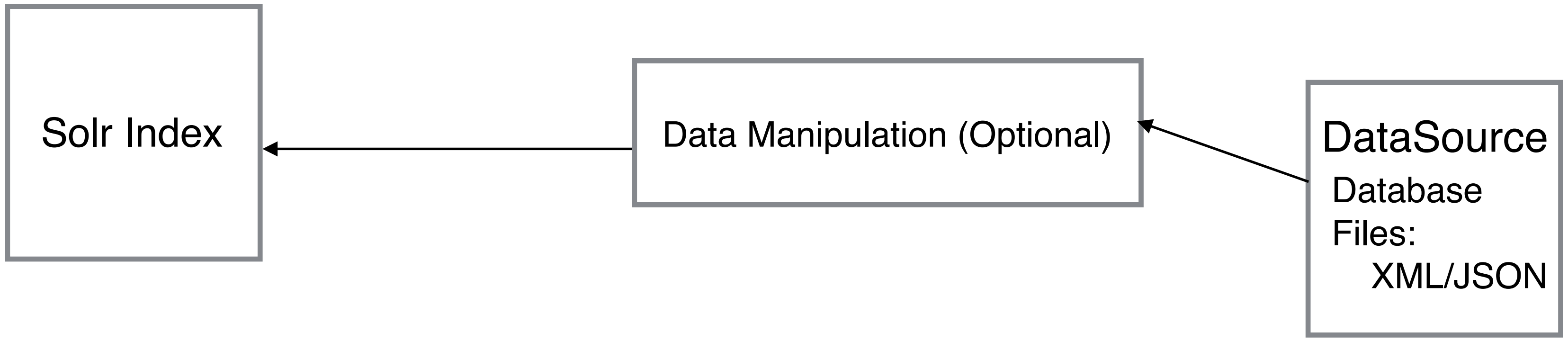
Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: History

- Built on Lucene
- Versions up with Lucene

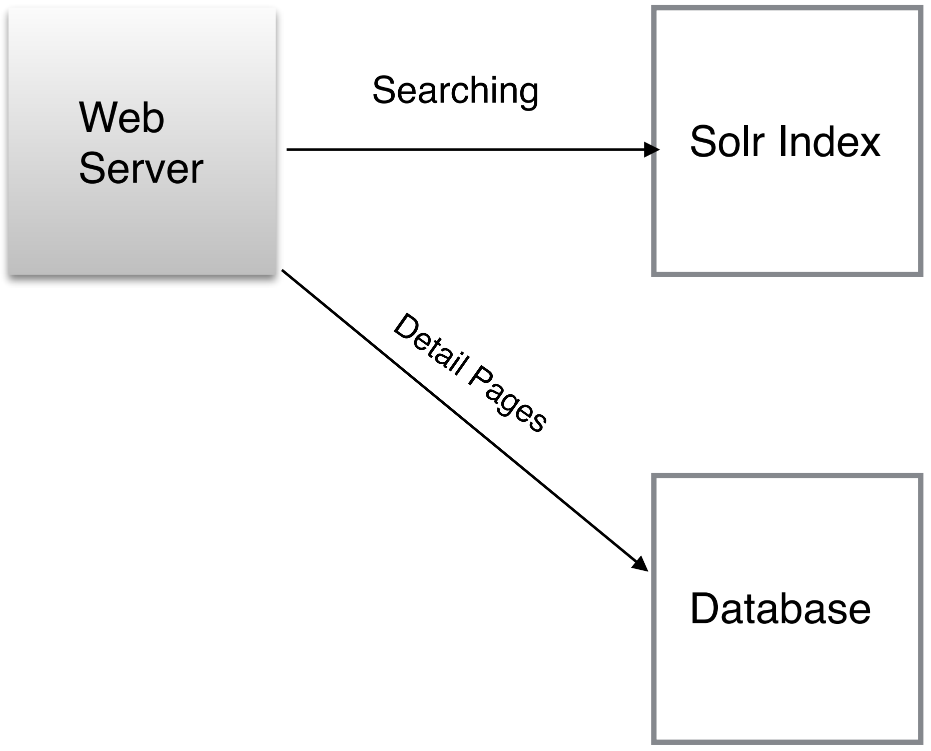
Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: Indexing



Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: Searching



Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

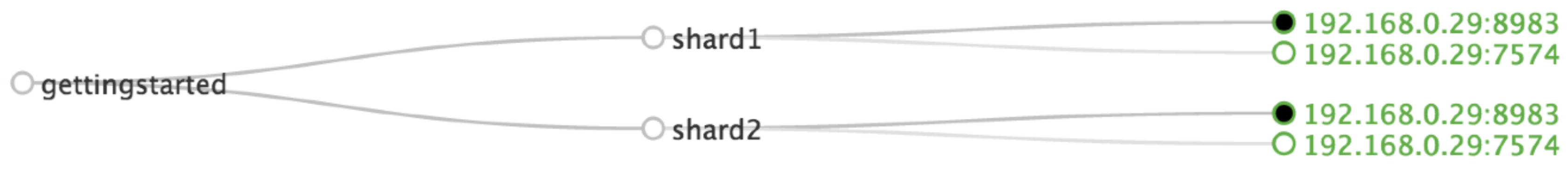
Solr: Master/Slave

- Main server responsible for indexing
- Slave servers
- Replication handled within Solr Software

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrCloud

- Built to simplify scalability
- Allows adding nodes to a cluster



Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr Admin

- Walkthrough

Part 1: Searching

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Searching: The Basics

- q=<field>:<value>
- Wildcards: * and ?
- Fuzzy searches: toy~ or toy~1
- Proximity: “toy story”~10
- Required: +toy
- Ranges:
  - Year:[1995 TO 1999] - Inclusive
  - Title:{Aladdin TO Boy} - Exclusive
- Boosting: ^10

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: Searching

- Try to find the movies with Woody Harrelson
- Find your favorite movie
- Get movies that were made in 2010 and 2011
- Return movies that were about disasters in 1999
- Find movies that have the word “world” and may have “danger”
- Find movies with a fuzzy search for fight
- Find movies with Deep in the title or the Plot
- Same as previous but boost documents with it in the title by 10

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Searching: Common Params

- sort DESC:ASC
- Field List: `fl=title,plot,year`
- Filter Query:
  - Applies filter without impacting the score
  - `fq=Year:1995&fq=actors:"Tom Hanks"`
- Write Type: `wt=json`
- Indent to pretty up the return: `indent=true`
- Start (offset): `start=10`
- Debug/ExplainOther

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

TF/IDF

- Term Frequency
  - Search: The Brown Cow
  - Documents more relevant where “the”, “brown” and “cow” occur the most
- Inverse Document Frequency
  - Adds weight to terms that are infrequent across all documents
  - “The” occurs many times across all documents
  - “brown” and “cow” get more weighting

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: Common Parameters

- Return json and indent it
- Get a list of movies with Title, Year, score and imdbRating for Adventure and show 50 records
- Sort the previous example on the highest rated and most recent
- Search for “batman” in plot and title but use fq to filter to the year 2013
- Use `debug=true` to parse through the matching of the previous example
- Play around with searching and see what else you can find

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: Facets

- Similar to group by
- Allows for facetted navigation
- Field must be indexed
- Facets on tokens not stored data
- `facet=true` Turns on the facet system
- `facet.field` Picks a field to facet on
- `facet.limit` Sets a max amount of facet values
- `facet.mincount` Sets minimum doc count for a facet

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: Facets

- `facet.prefix` Simply filters facets to prefix
- `facet.contains` (contains.ignorecase): Filters facets to ones that contain the string
- `facet.offset`: Starts the facets after the offset

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: Facet Ranges

- Useful for know ranges of values like ratings
- `facet.range`: Which field to do a range facet
- `facet.range.start (end)`: Start and end of range
- `facet.range.gap`: The amount of each range
- All properties can be set on a per field basis with `f.<fieldname>.<parameter>`

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr: Facet Intervals

- Useful for arbitrary groupings
- `facet.interval`: set the field
- `facet.interval.set`: set an interval
- * can be used as any for upper and lower bound
- Don't forget you can use `f.<fieldname>.facet.interval.set`

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Typical Facet Navigation

- Page Displays With Facets
- User clicks a link of the facet value
- New page displayed with the facet value converted to fq=<field>:<value>

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: Facets

- Facet on Year and genres
- Facet on actors but limit to only 5 facets
- Facet on Year. Limit them to years with more than 1000 movies in the data set
- Facet on all Baldwin brothers using contains
- Facet on Year in ranges of 5 years mincount of 1
- Interval Facet on imdbRating. 0 to 3, 3 to 5, 5 to 7, 8 to 9, 9 to 10

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Discussion: Searching

- How do you want to search?
- What bits of information do you need?
- What ways can you incorporate searching into your business?

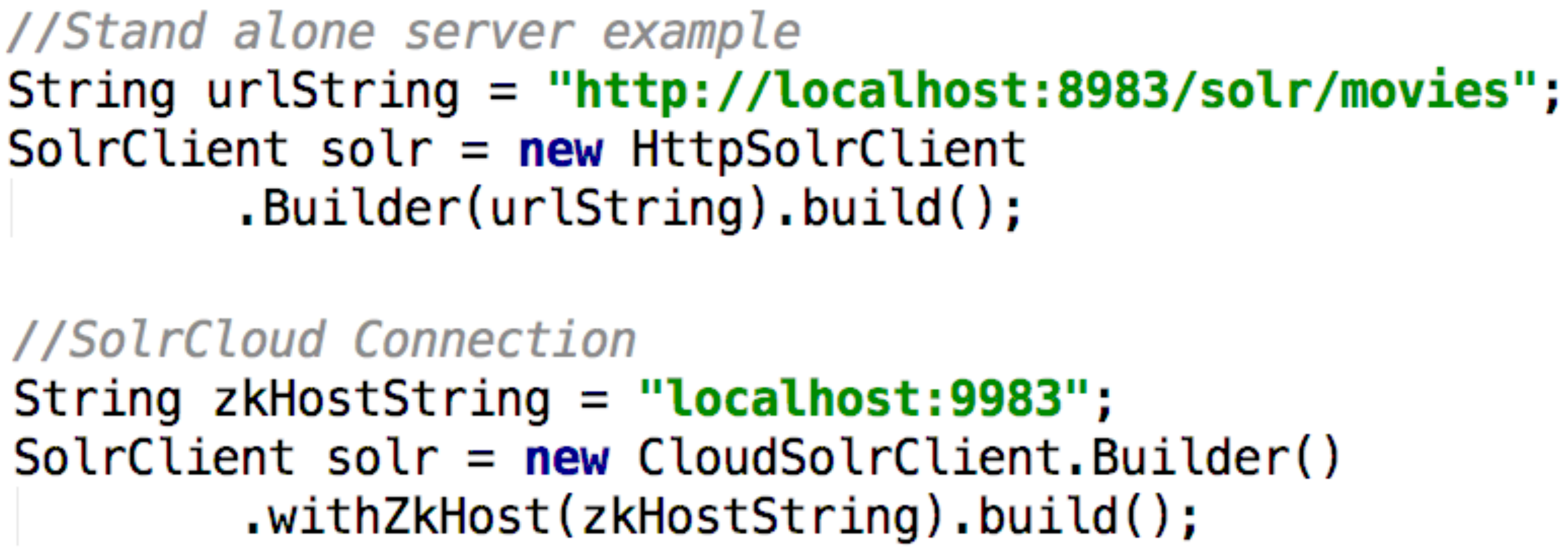
Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrJ

- Java client for solr
- Extremely limited documentation (1 page)
- Versions up with the version of solr
- Abstracts many of the communication details
- Adding documents easier with java based class
- Uses a fast java serialization

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrJ: Connecting

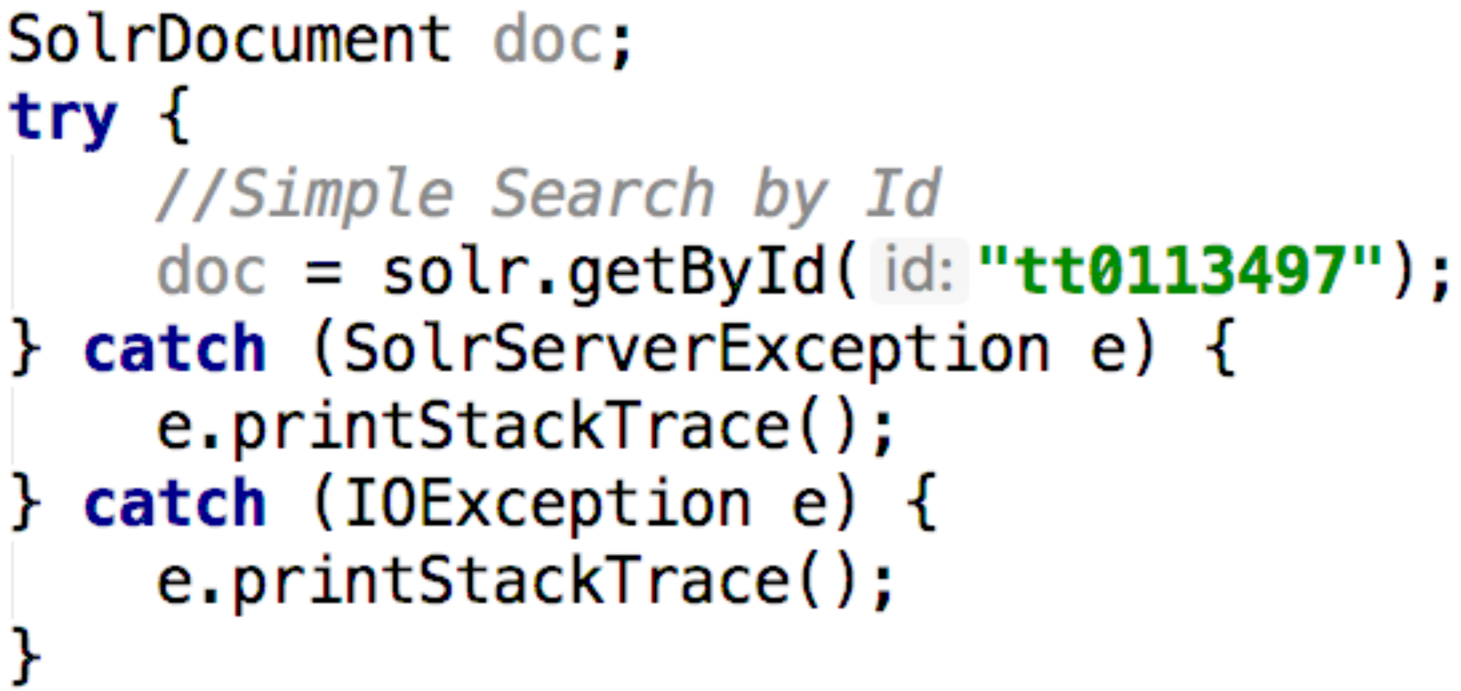


```java
//Stand alone server example
String urlString = "http://localhost:8983/solr/movies";
SolrClient solr = new HttpSolrClient
        .Builder(urlString).build();

//SolrCloud Connection
String zkHostString = "localhost:9983";
SolrClient solr = new CloudSolrClient.Builder()
        .withZkHost(zkHostString).build();
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

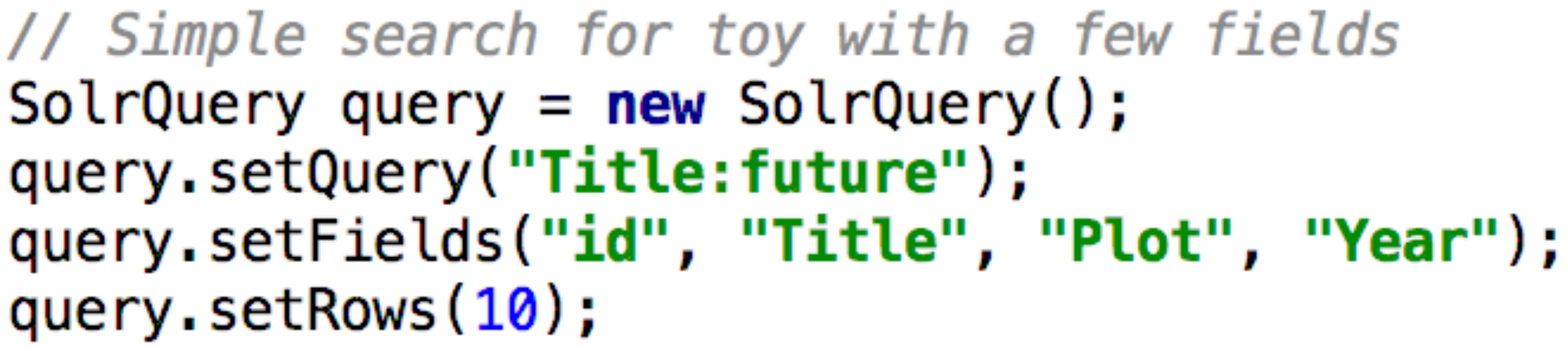
SolrJ: Searching By Id



```
SolrDocument doc;
try {
    //Simple Search by Id
    doc = solr.getById( id: "tt0113497");
} catch (SolrServerException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

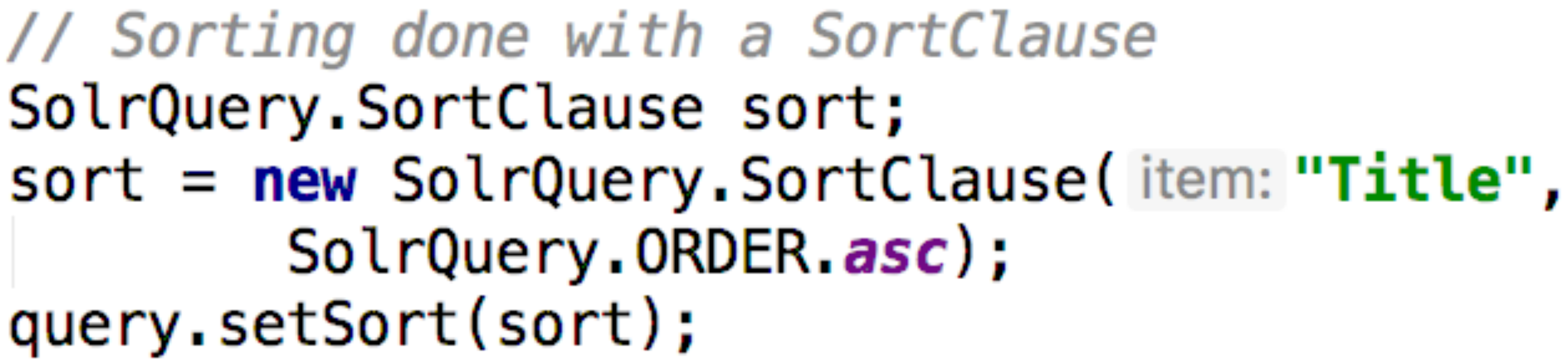
SolrJ: Search



```
// Simple search for toy with a few fields
SolrQuery query = new SolrQuery();
query.setQuery("Title:future");
query.setFields("id", "Title", "Plot", "Year");
query.setRows(10);
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

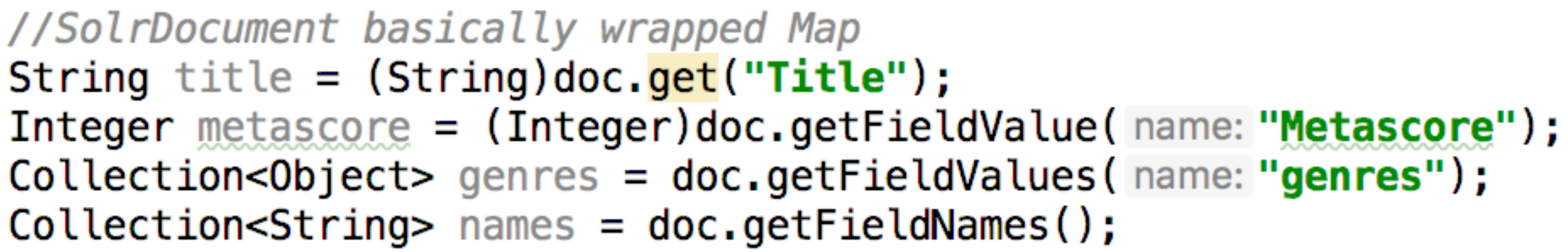
SolrJ: Sorting



```
// Sorting done with a SortClause
SolrQuery.SortClause sort;
sort = new SolrQuery.SortClause( item: "Title",
         SolrQuery.ORDER.asc);
query.setSort(sort);
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

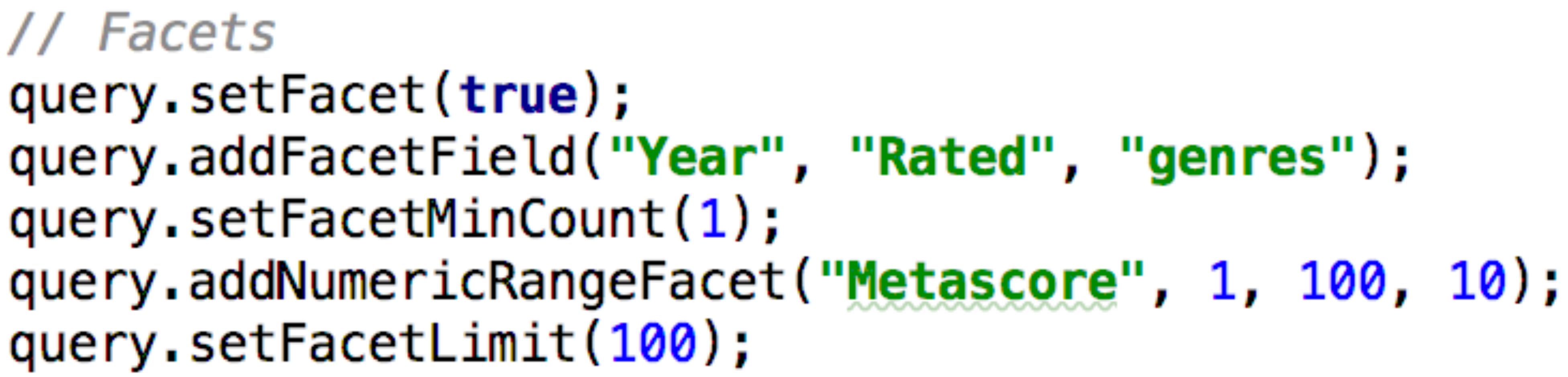
SolrJ: Search Response



```
//SolrDocument basically wrapped Map
String title = (String)doc.get("Title");
Integer metascore = (Integer)doc.getFieldValue( name: "Metascore");
Collection<Object> genres = doc.getFieldValues( name: "genres");
Collection<String> names = doc.getFieldNames();
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

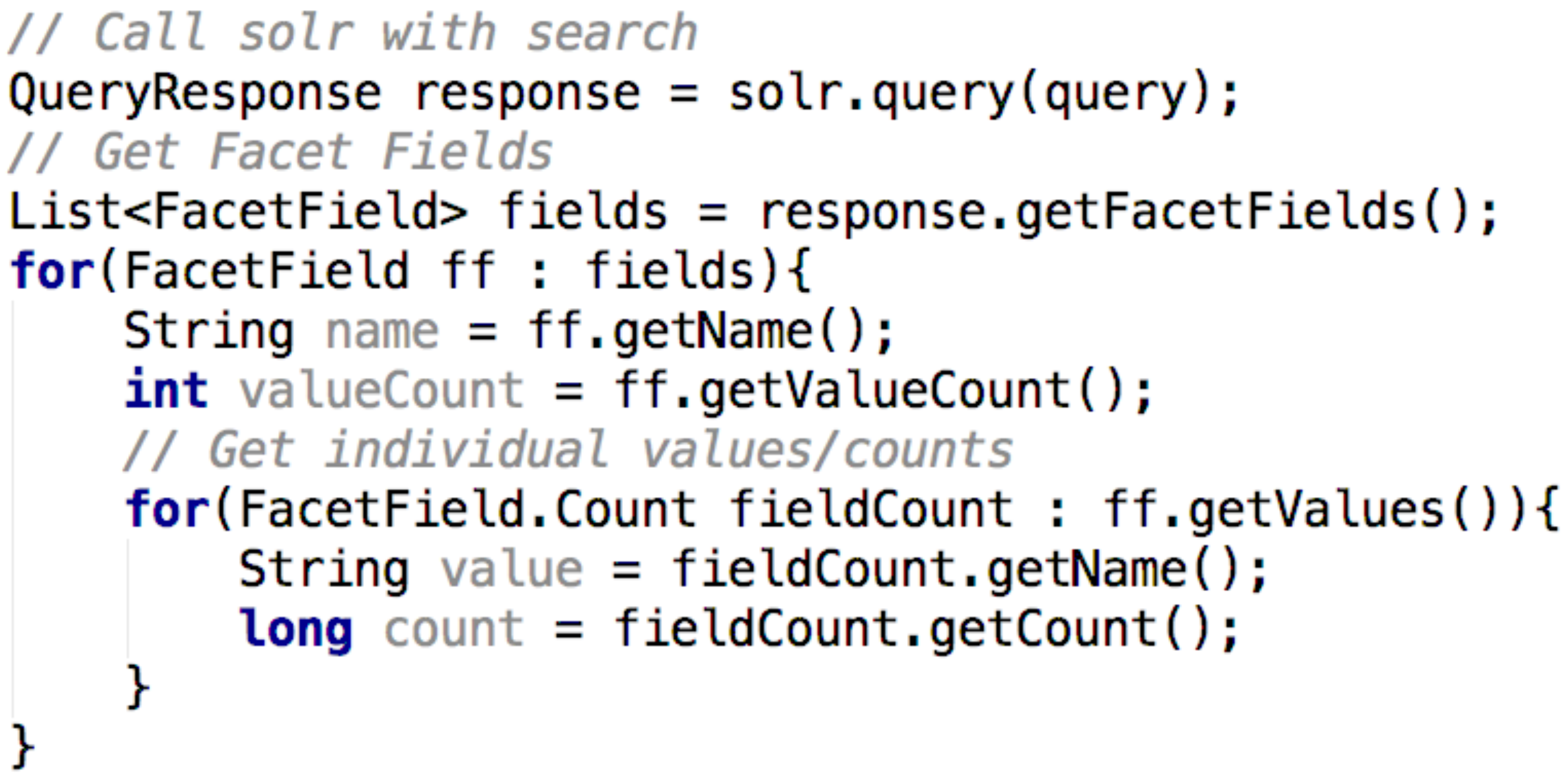
SolrJ: Facets



```java
// Facets
query.setFacet(true);
query.addFacetField("Year", "Rated", "genres");
query.setFacetMinCount(1);
query.addNumericRangeFacet("Metascore", 1, 100, 10);
query.setFacetLimit(100);
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrJ: Facet Response



```
// Call solr with search
QueryResponse response = solr.query(query);
// Get Facet Fields
List<FacetField> fields = response.getFacetFields();
for(FacetField ff : fields){
    String name = ff.getName();
    int valueCount = ff.getValueCount();
    // Get individual values/counts
    for(FacetField.Count fieldCount : ff.getValues()){
        String value = fieldCount.getName();
        long count = fieldCount.getCount();
    }
}
```

Searching Project

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

End of day survey

- http://bit.ly/solr3day4-17-17

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr Primer Day 2

Macy's

jeffnb@gmail.com

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Part 2: Schemas and Indexes

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Fields

- Fields store data
- Fields have a type
- Parameters
  - stored: values can be retrieved
  - multiValued: Can store multiple values
  - indexed: Can be searched/faceted
  - required: Document will error if value missing
  - type: The type of field
  - docValues: Orders values to be more efficient

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Basic FieldTypes

- Data types
- No tokenization
- string: most common
- int: solr.TrieIntField
- long: solr.TrieLongField
- float: solr.TrieFloatField
- double: solr.TrieDoubleField
- date: solr.TrieDateField

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: Basic Fields

- `bin/solr create_core -c movies -d <path>/movies-start`
- Open file: `server/solr/movies/conf/managed_schema`
- Find the id field in the file and add fields under. Add remaining fields with built in types from the following: Country, Rated, Language, imdbVotes, Type, Poster, Metascore, Year, actors, genres, directors, writers, Runtime, imdbID, Released, imdbRating

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Text FieldTypes

- The magic is here
- Many pre configured including text_general
- Have several steps in processing data
- Are the core power of solr

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Analyzers

- List of instructions to run
- 2 Times Run
  - Index: Documents going in
  - Query: Searching existing
- Contain steps to process data
- First Step Tokenizer

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Tokenizers

- Tokenizers take input and break it into “tokens”
  - The Quick Brown Fox Jumps Over the Lazy Dog
  - “The”, “Quick”, “Brown”, “Fox”, “Jumps”, “Over”, “the”, “Lazy”, “Dog”
- Can tokenize in a variety of ways including not breaking it up at all

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Tokenizer Overview

- Standard: Splits based on whitespace and punctuation
- Classic: Standard but doesn’t support Unicode Annex
- Keyword: Keeps entire input as a single token
- Letter: Tokens are strings of contiguous letters
- LowerCase: Delimits on non-letters and lowercases
- N-Gram: generates n-gram tokens
- Path Hierarchy: Replaces a delimiter with another value building up a path
- Regular expression: Tokenizes based on regular expression as a delimiter
- Url Email: Splits on white space and tokens preserving email and urls
- Whitespace: Splits on whitespace only

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Filters

- Alter tokens
- Produce more tokens
- Suppress tokens
- Dozens exist

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Filters: Heavily Used

- Lowercase Filter: Lowercases all tokens
- Stop Filter: Takes a file and removes all tokens that match words in the file
- Stemmers (porter/snowball): Remove word conjunctions and plurization
- Synonym Filter: Token matched to file and all synonyms added as tokens
- Pattern Replace Filter: Replaces a regular expression match with a string
- WordDelimiter Filter: splits tokens up further into words useful for breaking words apart or concatenating hyphens

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Filters: Less Common

- Managed Stopwords/Synonyms: Allows management via rest api
- N-Gram Filter: Breaks up token into small chunks
- Phonetic Filter: translates tokens into phonetic equivalent. Can replace or add
- Length Filter: Filters tokens by min/max length

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: Text Fields

- Create a field for Plot and Title
- Create a field type for Awards
- Create a field for awards using your own field type
- Use the analyzer admin to ensure these fields work

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

CopyField

- Shortcut to copy one field into another

```
<copyField source="<sourcefield>" dest="<destfield>"/>
```

- Useful for when facets and search are needed

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: CopyField

- Use copyField to copy the following to text:
  - actors, genres, directors, writers, Plot, Year, Title, Awards, language, Rated
- Create fields for (use a name like actors_text):
  - actors
  - genres
  - directors
  - writers
- Use copy field to copy the source fields above to the new _text versions

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Dynamic Fields

- DynamicField allows wild carding field names

```
<dynamicField name="*_en"  type="text_en"
indexed="true"  stored="true" multiValued="true"/>
```

- Any field with _en at the end automatically populates into that field
- Can be referenced directly

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: DynamicFields

- Change all _text fields to simply copy into _en fields
- Explore a few of the other dynamic fields

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Discussion

- Pick a feature Content, Product, Reviews
- What are major fields needed?
- What types are the fields?
- What analysis could be used?

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Schema API

- Allows for schema changes dynamically
- Allows for adding, changing, deleting fields and field types
- Caveat: Changing a field does not change data
- Main entry point:
  - `http://<host:port>/solr/<core>/schema`
- Reading fields or a specific field:
  - `/solr/<core>/schema/fields/`
  - `/solr/<core>/schema/fields/<fieldname>`

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: Schema API

- Retrieve the list of the defined fields for the index
- Retrieve the Plot field specifically
- Add a new field RTReview (int) for storing rotten tomatoes
- Add a string field to store tags
- Add a field type to search tags should trim, lowercase, and stem them
- Create a copy directive to copy the first tags field into the new field type

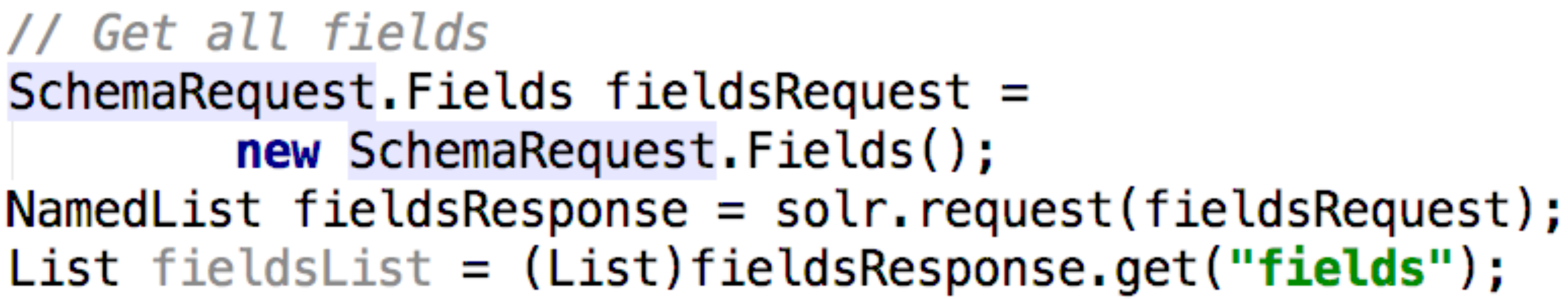
Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrJ: Schema

- Package Exists
- No documentation page
- Uses solr.request.schema
- Use a series of request objects to change schema
- Very similar to REST API from last section
- Link in the git repo

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

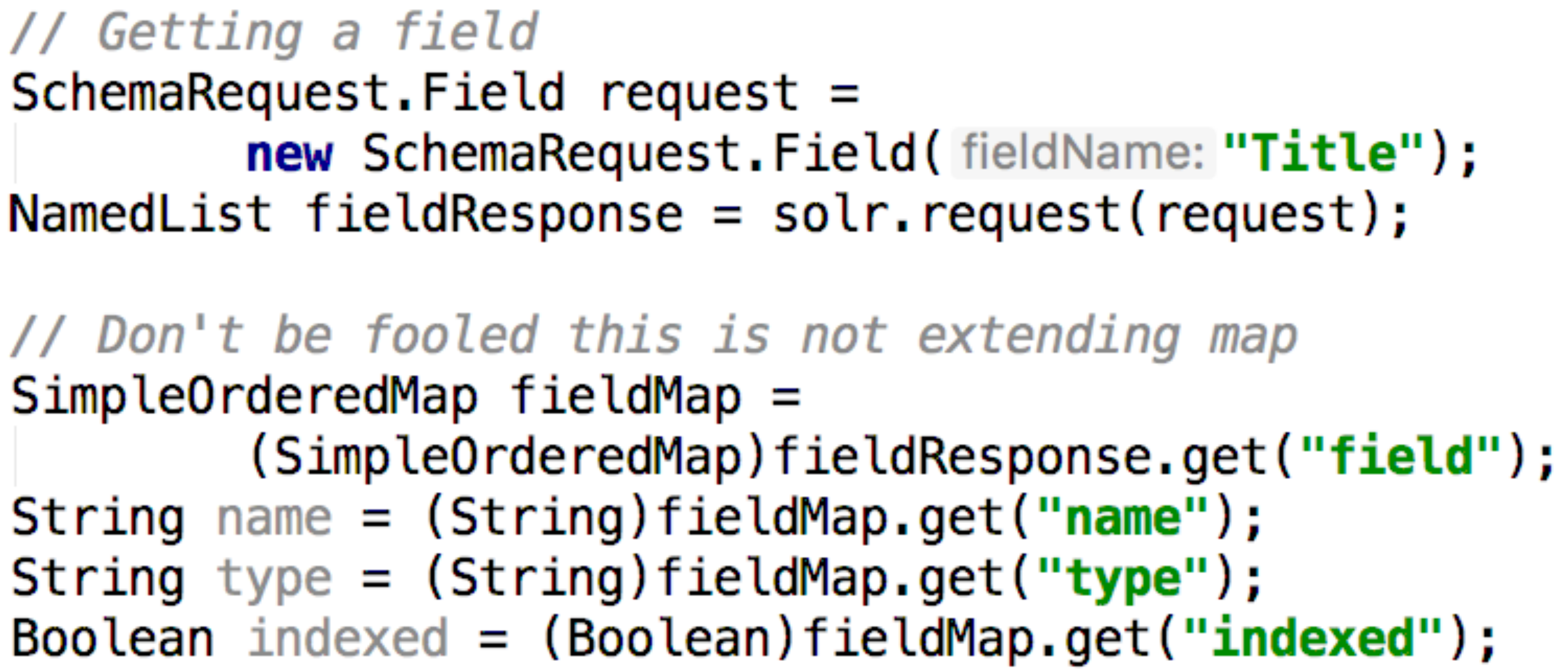
SolrJ: List Fields



```
// Get all fields
SchemaRequest.Fields fieldsRequest =
        new SchemaRequest.Fields();
NamedList fieldsResponse = solr.request(fieldsRequest);
List fieldsList = (List)fieldsResponse.get("fields");
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrJ: Field Schema

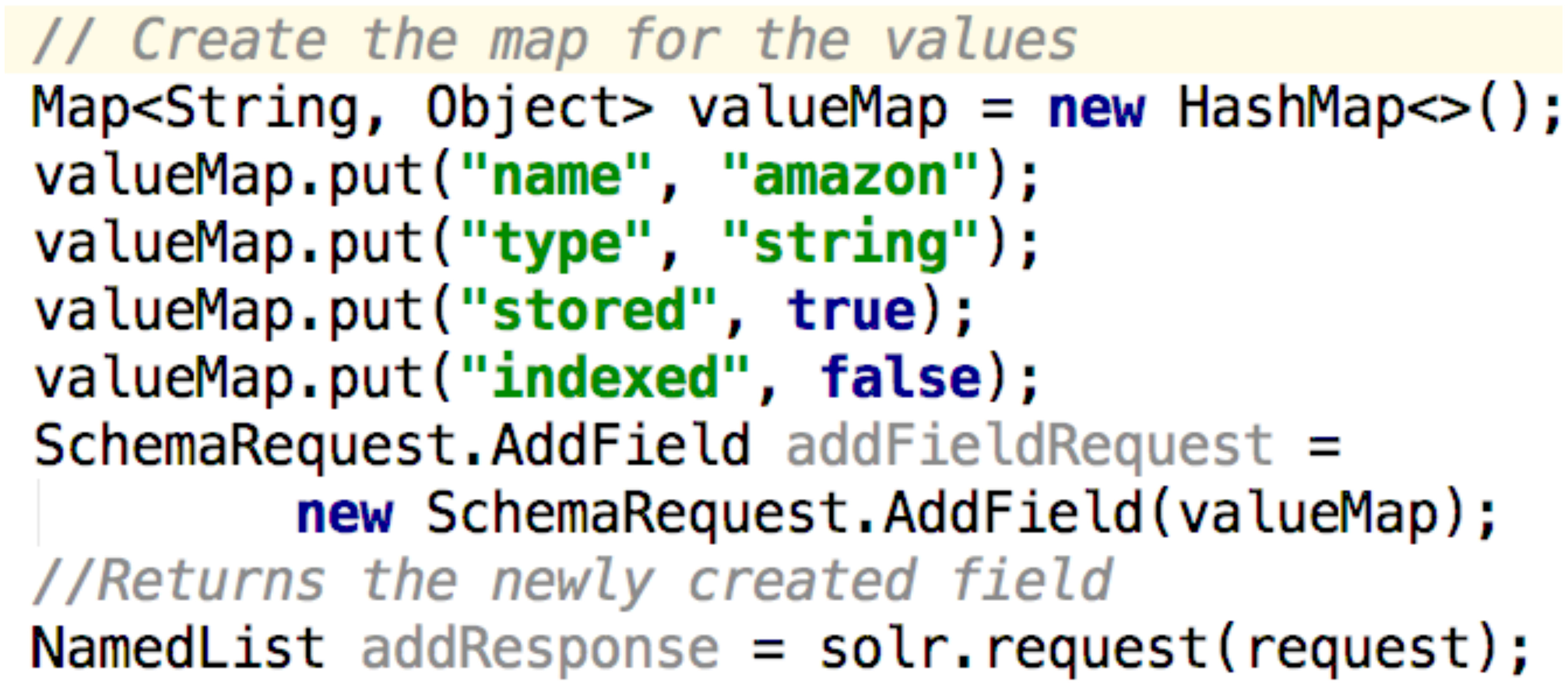


```java
// Getting a field
SchemaRequest.Field request =
        new SchemaRequest.Field( fieldName: "Title");
NamedList fieldResponse = solr.request(request);

// Don't be fooled this is not extending map
SimpleOrderedMap fieldMap =
        (SimpleOrderedMap)fieldResponse.get("field");
String name = (String)fieldMap.get("name");
String type = (String)fieldMap.get("type");
Boolean indexed = (Boolean)fieldMap.get("indexed");
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrJ: Add Field



```java
// Create the map for the values
Map<String, Object> valueMap = new HashMap<>();
valueMap.put("name", "amazon");
valueMap.put("type", "string");
valueMap.put("stored", true);
valueMap.put("indexed", false);
SchemaRequest.AddField addFieldRequest =
        new SchemaRequest.AddField(valueMap);
//Returns the newly created field
NamedList addResponse = solr.request(request);
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Schemaless

- Solr can run in schemaless (guessing) mode
- Data will determine the schema
- New fields are added automatically

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Part 3: Indexing

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Indexing

- Process to add to the index
- Common ways:
  - `bin/post` file
  - Post request to update handler
  - Data Import Handler

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Deleting

- Leaves document in index hidden
- Posting with body
  - {"delete": { "id":"12345" }}
  - {"delete": { “query”:”Title:Toy Story" }}

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Committing

- Adding records keeps them in staging
- Commit makes the records update the index
- Similar to database transactions
- bin/post commits automatically
- Sending a commit:
  - http://localhost:8983/solr/movies/update?commit=true

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Commit: Soft vs Hard

- Hard commits write to disk
- Soft commits:
  - Makes changes visible quicker
  - Changes not written
  - Crashes can lose data

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Optimizing

- Optimization cleans up the index
- Reduces amount of index files
- Commit does not remove deleted docs
- Optimize makes the index nice and clean
- Can speed up search times

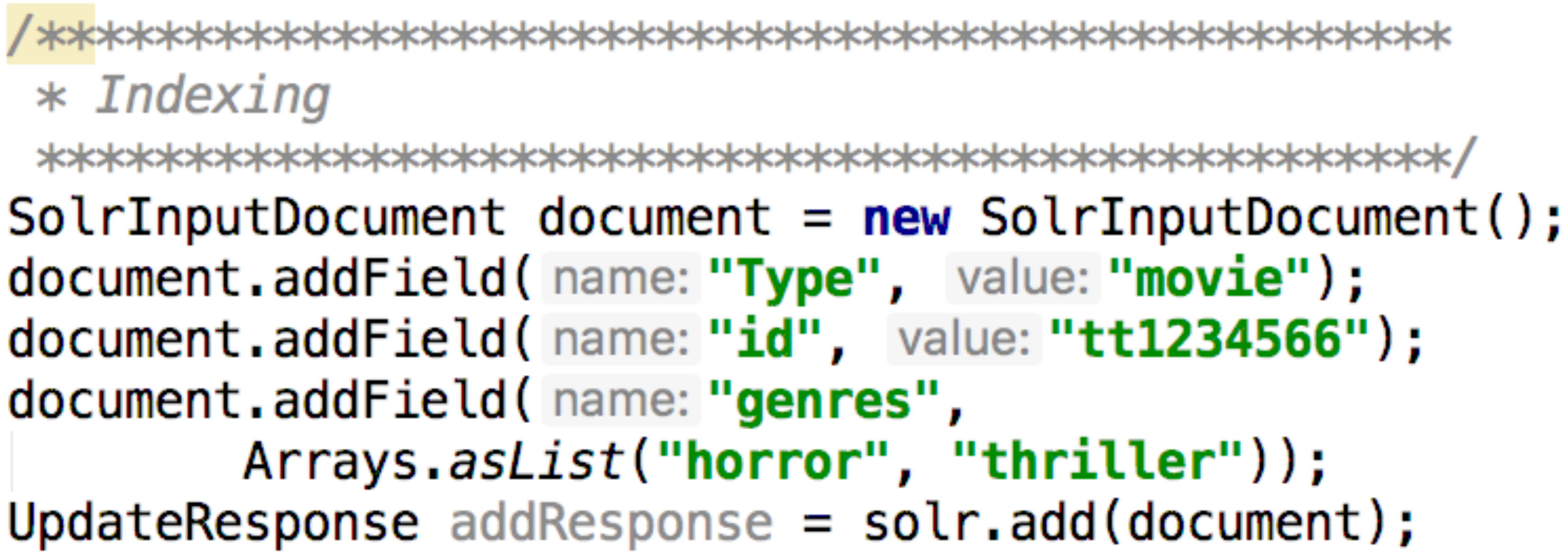
Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: Load the Data

- Moment of Truth
  - bin/post -c movies <path>/movie_data_cleaned.json
- Fix errors or ask questions
- Delete your least favorite movie
- Run an optimization if num docs and max docs are different

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

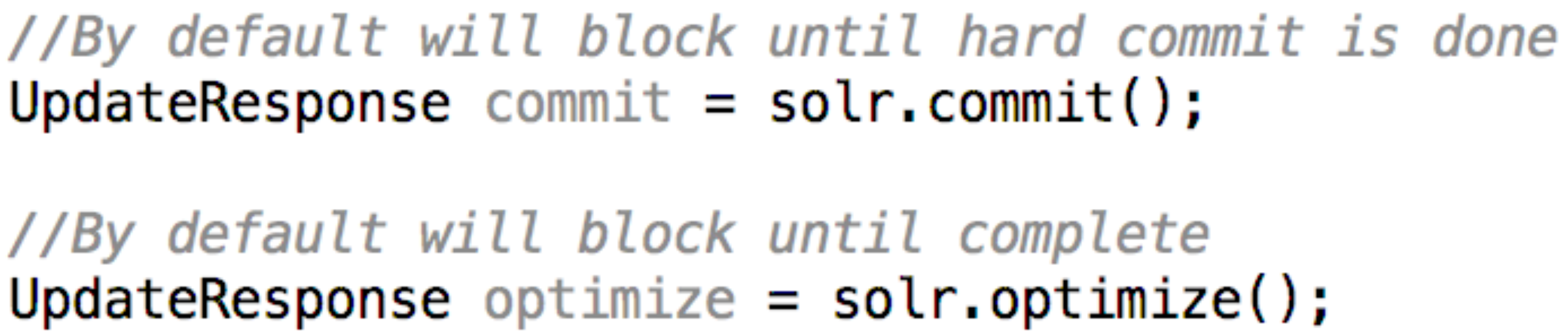
SolrJ: Adding Documents



```
/************************************************
 * Indexing
 ************************************************/
SolrInputDocument document = new SolrInputDocument();
document.addField( name: "Type",  value: "movie");
document.addField( name: "id",  value: "tt1234566");
document.addField( name: "genres",
        Arrays.asList("horror", "thriller"));
UpdateResponse addResponse = solr.add(document);
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrJ: Commit/Optimize



```
//By default will block until hard commit is done
UpdateResponse commit = solr.commit();

//By default will block until complete
UpdateResponse optimize = solr.optimize();
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Project: Building A Core

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Solr Primer
Day 3

Macy's

jeffnb@gmail.com

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Part 4: Configurations

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrConfig.xml

- Defines:
  - All Handlers
  - Commit Behavior
  - Caching
  - Default behaviors
  - Search Components installed
  - Schemaless configuration
  - Default and environmental variables
  - Listeners

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

AutoCommit

- Allows commits to be run automatically
- Useful when trickle changes come in
- Helps avoid many small commits

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

AutoSoftCommit

- Commits will be visible
- Commits not written to disk
- Useful with AutoCommit
  - AutoCommit at longer intervals
  - AutoSoftCommit at short intervals

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Caches

- Query
  - Stores document ids for common queries
  - Cuts down on multiple index calls to find docs
- Filter
  - Stores document ids for common filter queries
  - Speeds up common filters considerably
- Document
  - Stores commonly requested fields
  - Helps speed up document retrieval

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Searchers

- Similar to a thread
- Searchers process the request
- Searchers have their own caches
- Have events:
  - Trigger when the first searcher starts
  - Trigger when new searchers start
  - Useful for warming with common queries

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: Commits and Caches

- Set up auto commit in the movies solrconfig file
  - Commit every 5 minutes
  - Soft commit every minute
  - Ensure new searcher is opened on commits
- Experiment tuning the caches
  - Enable a filter cache which auto warms with 200 entries
  - Enable a query cache with 20 entries auto warmed
  - Look at the admin Plugins->Cache and checkout the cache levels. Try queries and watch them change
- Run 3 queries on startup when searcher is warmed

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Search Handlers

- End points for querying
- Several come with solrconfig
- Built in ones can be overridden
- New ones can be created (edismax from earlier)
- Specifies all behaviors
- `class="solr.SearchHandler"`

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Search Handlers

- Defaults for queries
  - `<lst name="defaults">`
  - Specifies any query defaults
  - Useful for fl, rows and wt
- Appends
  - `<lst name=“appends">`
  - Allows values to queries
  - User cannot override
  - Example: `<str name="fq">inStock:true</str>`

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Search Handlers [cont]

- InitiParams
  - Specifies a set of parameters across specified handlers
  - Overridden by individual handlers
- Invariants
  - `<lst name="invariants">`
  - Used to limit options on queries
  - Example: `<str name=“facet.field">genres</str>`
  - Once specified no changes are allowed query time for that name

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: Search Handler

- Create a new search handler for Action movies
- Specify these defaults:
  - fl with Title, Plot, Year, genres, imdbRating
  - rows of 50
  - wt of json
  - turn on indent
- Append genres_en:Action to all queries
- Allow faceting on Year and actors

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

eDismax

- Advantages
  - Intended to not throw errors
  - Syntax more forgiving
  - Can search over many fields
  - Useful for user searches
- Extremely powerful
- Flexible for different needs
- Run with: `defType=edismax`

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

eDismax parameters

- q - Term to search for
- qf
  - Fields to use to search with boosts

```
<str name="qf">
   Title^10 Plot^5
</str>
```

- mm - Minimum amount of terms to match
- pf
  - Boosts based on fields whole phrases appear in.
  - Same syntax as qf
- bq
  - Boosts based on a query
  - bq=genres:action^3.0

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: eDismax Handler

- Create a new search handler “edismax”
- For search fields:
  - Title and Plot are the highest boosted
  - actors, directors and writers have a lower boost
  - Year and Language are searched but not boosted
- All terms should match
- Boost (pf) Title and plot again
- Rows should be 30
- Fields returned should be Title, Plot and Year

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Discussion

- Now that you have a grasp on query options
- What fields do you think are most important?
- What would you want to return by default?

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Config API

- Rest API exists
- Doesn’t appear to be supported in SolrJ
- Can administer the config through the api
- Stand alone mode would only replicate with a master server

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Config API: Reading

- Entire Config
  - http://localhost:8983/solr/demo/config/
- Specific Section
  - http://localhost:8983/solr/demo/config/requestHandler
- Specific Component
  - http://localhost:8983/solr/demo/config/requestHandler?componentName=/select

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Config API: Common Commands

- Updating caches
- Changing auto commit values
- JMX changes

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Config API: General Purpose

- Request Handlers
- Search Components
- Init params

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Part 4: Optimization

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Overview

- Solr is big
- No silver bullet
- Should be based on user behavior
- Dependent on your architecture

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

System Level

- Memory
  - Do not give Solr all the memory
  - At least half should be left for the OS
  - Turn up OS file cache to get index in memory
  - Make sure open file limit is high enough
- Hardware
  - Master/Slave servers work much better on bare metal
  - SSDs are great speed improvements if possible
  - Ram can boost performance

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Caching

- Look at the logs
- Warm searchers with common queries
- Autowarm when you can
- Start with defaults and increase or decrease levels
- Balance memory with caches
- Use the cache admin to get hit rates

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

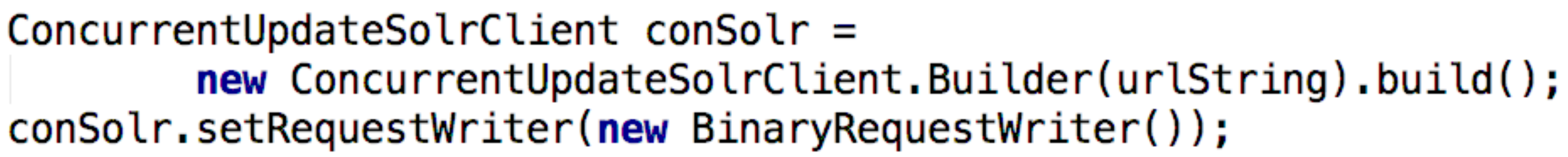
Distributed Indexes

- Indexes slow with size
- Threshold highly variable (millions of records)
- Sharding: breaking up index
- Solr HIGHLY recommends not sharding with M/S setup
- If index becomes too large move to SolrCloud

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Document Adding

- Change Update Format
  - Default XML
  - Change to binary
  - Only works after 3.1
- Concurrent updates.
  - Uses multiple HTTP connections
  - Speeds up large indexing



```
ConcurrentUpdateSolrClient conSolr =
        new ConcurrentUpdateSolrClient.Builder(urlString).build();
conSolr.setRequestWriter(new BinaryRequestWriter());
```

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Commits

- Commits slow throughput
- Searches are expensive to open
- Be practical about real time data needs
- Use soft commits to supplement hard commits
- Transaction logs will grow out of control without hard commits

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Optimizes

- Slow throughput down
- Optimizing creates new files
- Optimizing merges files making index smaller
- Overall better performance when optimized
- Better run at low times

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Replication

- Replication can cause new searchers
- Will temporarily slow searches down
- Understand real time needs

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Garbage Collection

- Larger Solr instances
- Can stop the entire server for seconds
- Older versions of GC “stop the world”
- CMS better for 1.6
- G1 better for 1.7+

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Part 5: SolrCloud

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

SolrCloud Overview

- Grew out of scalability troubles
- Closest to ElasticSearch methods
- Handles replication and sharding
- Uses ZooKeeper to manage nodes
- Allows new servers to plug in with ease
- Has collections not cores

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Lab: SolrCloud

- Use bin/solr to create a movies collection
- Use 2 shards and 2 replicas
- Import movies data
- Explore diagrams in admin interface

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Part 6: NLP and Solr

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

What is NLP

- Natural Language Processing

Natural language processing (**NLP**) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages and, in particular, concerned with programming computers to fruitfully process large natural language corpora.

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

What is NLP

- A way to help computers understand human language
- Instead of looking at each word individually (solr)
- Takes the context of a sentence or document

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Human Language

- Extremely difficult
- New words constantly that don't have any context in existing models
- Multiple meanings are a problem: Spoke (speaking) vs Spoke (in a bike wheel)
- Modifying words can change the feel of a sentence "It was a hardly fun" vs "It was incredibly fun"
- Only 30% of communication is based on the words said

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

NLP: Common Tasks

- Sentence Breaking
- Tokenization
- Tagging
- Lemmatization/Stemming
- Name Entity Extraction

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

NLP: Sentence Breaking

- Splitting input on sentence boundaries
- Can be tricky with punctuation … vs .
- Important first step
- NLP is all about context
- Sentences are a discrete idea.

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

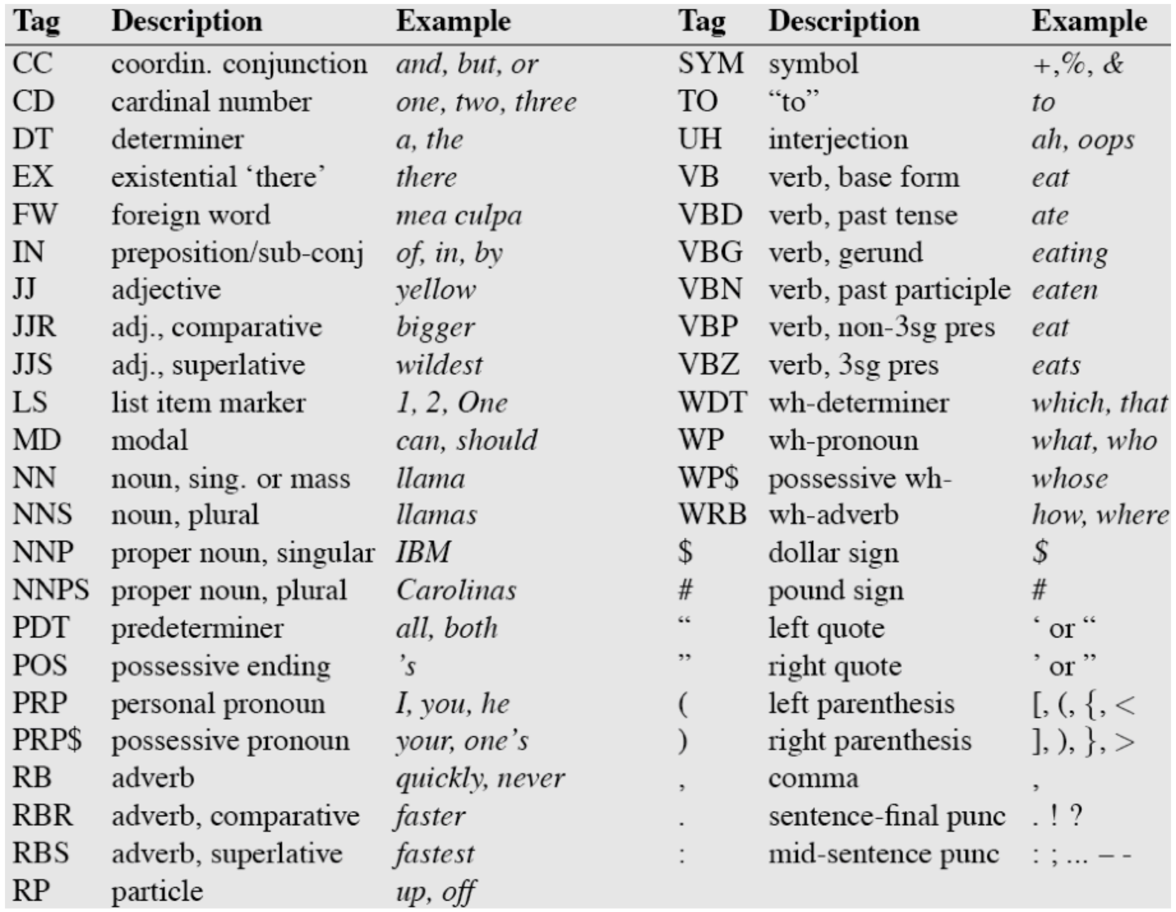
NLP: Tokenization

- Similar to Solr ideas of tokenization
- Can take many forms
- Usually special versions of white space
- Keeps punctuation in the list of tokens

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

NLP: Tagging

- Processes sentence and tokens
- Tags each token with a type of speech
- Spoke => Spoke/VBD
- Running => Running/VBG



Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	“to”	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential ‘there’	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	$
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*’s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one’s*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

NLP: Lemmatization

- Similar to stemmers
- Uses tags to determine roots
- Root is always a full word
- Example:
  - Speaking/VBG and Spoke/VB => speak

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

NLP: Named Entity Extraction

- Pulls out names from input
- Can be made to identify people vs places
- Interesting usage of process

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Machine Learning

- How natural language processing is based
- Take training data and run it through model
- Model “learns” from the data to associate certain desired outcomes
- Testing data then run through system to ensure outcome correct

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

OpenNLP

- Apache Project
- Handles common Tasks
- Has many different models
- Models are downloaded separately
- Stand alone project
- Java available with Maven include

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

OpenNLP: Pros and Cons

- Pros:
  - Open source
  - Rudimentary connection to Solr
  - Flexible with including models
- Cons:
  - Small community
  - Not particularly active
  - Models aren't included due to licensing

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

OpenNLP and Solr

- No standard integration
- Famous Lucene-2899 Patch
  - Written 6 years ago
  - Updated last year
  - Missing any way to process named entities
  - Memory Inefficient
- Download libs and include in Solr package
- Allows for common steps mentioned

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

NLP: Considerations

- OpenNLP is not the biggest or best
  - NTK - Python
  - CoreNLP - Stanford
- NLP is good for language not searches
  - Lemmas and Tagging break down in keyword searching
  - “Search Salad” is very difficult to understand
- Lemmatization struggles with new words
- Just a start. Extra work needed to effectively use

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com

Thank you

Please fill out the surveys

Copyright 2017 DevelopIntelligence LLC
http://www.DevelopIntelligence.com