Item 291
git_comments:

1. make a deep copy
2. default constructor
3. default constructor
4. default constructor
5. * * Utility that converts DMOZ * RDF into a flat file of URLs to be injected.
6. **comment:** do nothing as this fields are optional TODO Make configurable?
   label: requirement
7. * * 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.
8. default constructor
9. default constructor
10. default constructor

git_commits:

1. **summary:** Merge pull request #295 from lewismc/NUTCH-2516
   message: Merge pull request #295 from lewismc/NUTCH-2516 NUTCH-2516 Hadoop imports use wildcards

github_issues:

github_issues_comments:

github_pulls:

1. **title:** NUTCH-2516 Hadoop imports use wildcards
   body: Large but simple patch which addresses https://issues.apache.org/jira/projects/NUTCH/issues/NUTCH-2516

github_pulls_comments:

1. I would like to commit by EoB today if possible. We can then shift priority to #299 I will that through with @kpm1985 Any comments folks?

github_pulls_reviews:

1. What about ivy/ivy-2.4.0.jar which has been committed in f9981d4? It's supposed to be "installed" by `ant ivy-download`.
2. Thanks @sebastian-nagel I've updated the PR to also prevent ```ivy-2.4.0.jar``` from being added to SCM.
3. **body:** +1 Maybe also remove this file to avoid confusions?
   label: code-design
4. and please go forward and commit, thanks!

jira_issues:

1. **summary:** Hadoop imports use wildcards
   description: Right now the Hadoop imports use wildcards all over the place. We wanted to address this during NUTCH-2375 but didn't get around to it. We should address it in a new issue as it is still important.

jira_issues_comments:

1. lewismc opened a new pull request #295: NUTCH-2516 Hadoop imports use wildcards URL: https://github.com/apache/nutch/pull/295 Large but simple patch which addresses https://issues.apache.org/jira/projects/NUTCH/issues/NUTCH-2516 ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org

2. lewismc commented on issue #295: NUTCH-2516 Hadoop imports use wildcards URL: https://github.com/apache/nutch/pull/295#issuecomment-376265552 I would like to commit by EoB today if possible. We can then shift priority to #299 I will that through with @kpm1985 Any comments folks? ---------- ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
3. sebastian-nagel commented on a change in pull request #295: NUTCH-2516 Hadoop imports use wildcards URL: https://github.com/apache/nutch/pull/295#discussion_r177200055 ########## File path: .gitignore ########## @@ -6,3 +6,8 @@ build/ runtime/ logs/ /bin/ +.classpath +.naivebayes-model.crc +.project +ivy/ivy-2.3.0.jar Review comment: What about ivy/ivy-2.4.0.jar which has been committed in f9981d4? It's supposed to be "installed" by `ant ivy-download`. ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
4. lewismc commented on a change in pull request #295: NUTCH-2516 Hadoop imports use wildcards URL: https://github.com/apache/nutch/pull/295#discussion_r177386301 ########## File path: .gitignore ########## @@ -6,3 +6,8 @@ build/ runtime/ logs/ /bin/ +.classpath +.naivebayes-model.crc +.project +ivy/ivy-2.3.0.jar Review comment: Thanks @sebastian-nagel I've updated the PR to also prevent ```ivy-2.4.0.jar``` from being added to SCM. ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
5. sebastian-nagel commented on a change in pull request #295: NUTCH-2516 Hadoop imports use wildcards URL: https://github.com/apache/nutch/pull/295#discussion_r177395421 ########## File path: .gitignore ########## @@ -6,3 +6,8 @@ build/ runtime/ logs/ /bin/ +.classpath +.naivebayes-model.crc +.project +ivy/ivy-2.3.0.jar Review comment: +1 Maybe also remove this file to avoid confusions? ---------- ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
6. sebastian-nagel commented on a change in pull request #295: NUTCH-2516 Hadoop imports use wildcards URL: https://github.com/apache/nutch/pull/295#discussion_r177395523 ########## File path: .gitignore ########## @@ -6,3 +6,8 @@ build/ runtime/ logs/ /bin/ +.classpath +.naivebayes-model.crc +.project +ivy/ivy-2.3.0.jar Review comment: and please go forward and commit, thanks! ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org
7. lewismc closed pull request #295: NUTCH-2516 Hadoop imports use wildcards URL: https://github.com/apache/nutch/pull/295 This is a PR merged from a forked repository. As GitHub hides the original diff on merge, it is displayed below for the sake of provenance: As this is a foreign pull request (from a fork), the diff is supplied below (as it won't show otherwise due to GitHub magic): diff --git a/.gitignore b/.gitignore index 5b3c68730..f44d4e705 100644 --- a/.gitignore +++ b/.gitignore @@ -6,3 +6,9 @@ build/ runtime/ logs/ /bin/ +.classpath +.naivebayes-model.crc +.project +ivy/ivy-2.3.0.jar +ivy/ivy-2.4.0.jar +naivebayes-model diff --git a/ivy/ivy-2.4.0.jar b/ivy/ivy-2.4.0.jar deleted file mode 100644 index 14ff88e26..000000000 Binary files a/ivy/ivy-2.4.0.jar and /dev/null differ diff --git a/src/java/org/apache/nutch/crawl/CrawlDatum.java b/src/java/org/apache/nutch/crawl/CrawlDatum.java index 1facf0a65..b50d9c92d 100644 --- a/src/java/org/apache/nutch/crawl/CrawlDatum.java +++ b/src/java/org/apache/nutch/crawl/CrawlDatum.java @@ -17,17 +17,27 @@ package org.apache.nutch.crawl; -import java.io.*; -import java.util.*; +import java.io.DataInput; +import java.io.DataOutput; +import java.io.IOException; +import java.util.Date; +import java.util.HashMap; +import java.util.HashSet; +import java.util.Map; import java.util.Map.Entry; import org.apache.commons.jexl2.JexlContext; import org.apache.commons.jexl2.Expression; import org.apache.commons.jexl2.MapContext; - -import org.apache.hadoop.io.*; -import org.apache.nutch.util.*; +import org.apache.hadoop.io.FloatWritable; +import org.apache.hadoop.io.IntWritable; +import org.apache.hadoop.io.Text; +import org.apache.hadoop.io.VersionMismatchException; +import org.apache.hadoop.io.Writable; +import org.apache.hadoop.io.WritableComparable; +import org.apache.hadoop.io.WritableComparator; import org.apache.nutch.protocol.ProtocolStatus; +import org.apache.nutch.util.StringUtil; /* The crawl state of a url. */ public class CrawlDatum implements WritableComparable<CrawlDatum>, Cloneable { @@ -36,7 +46,7 @@ public static final String FETCH_DIR_NAME = "crawl_fetch"; public static final String PARSE_DIR_NAME = "crawl_parse"; - private final static byte CUR_VERSION = 7; + private static final byte CUR_VERSION = 7; /** Compatibility values for on-the-fly conversion from versions < 5. */ private static final byte OLD_STATUS_SIGNATURE = 0; @@ -371,10 +381,8 @@ public void set(CrawlDatum that) { this.modifiedTime = that.modifiedTime; this.signature = that.signature; if (that.metaData != null) { - this.metaData = new org.apache.hadoop.io.MapWritable(that.metaData); // make - // a - // deep - // copy + // make a deep copy + this.metaData = new org.apache.hadoop.io.MapWritable(that.metaData); } else { this.metaData = null; } diff --git a/src/java/org/apache/nutch/crawl/CrawlDb.java b/src/java/org/apache/nutch/crawl/CrawlDb.java index a6d6a952e..a5455099d 100644 --- a/src/java/org/apache/nutch/crawl/CrawlDb.java +++ b/src/java/org/apache/nutch/crawl/CrawlDb.java @@ -17,24 +17,32 @@ package org.apache.nutch.crawl; -import java.io.*; -import java.util.*; +import java.io.File; +import java.io.IOException; +import java.io.FileOutputStream; +import java.io.OutputStreamWriter; +import java.io.BufferedWriter; +import java.io.FileNotFoundException; +import java.util.Random; +import java.util.Collection; +import java.util.Iterator; import java.text.*; import org.apache.hadoop.fs.Path; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.conf.Configured; import org.apache.hadoop.util.Tool; import org.apache.hadoop.util.ToolRunner; -import org.apache.hadoop.mapred.*; +import org.apache.hadoop.mapred.JobClient; +import org.apache.hadoop.mapred.JobConf; import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.fs.FileStatus; import org.apache.hadoop.fs.FileUtil; import org.apache.hadoop.fs.LockUtil; import org.apache.hadoop.fs.PathFilter; -import org.apache.nutch.util.*; +import org.apache.nutch.util.NutchConfiguration; +import org.apache.nutch.util.NutchJob; +import org.apache.nutch.util.NutchTool; import org.apache.nutch.util.lib.MurmurHash; -import org.apache.nutch.net.*; -import org.apache.nutch.metadata.Nutch; import org.apache.commons.jexl2.Expression; import org.apache.commons.jexl2.JexlEngine; import org.slf4j.Logger;

java.io.*; +import java.io.File; +import java.io.IOException; import java.lang.invoke.MethodHandles; import java.text.SimpleDateFormat; -import java.util.*; +import java.util.ArrayList; +import java.util.Arrays; +import java.util.HashMap; +import java.util.HashSet; +import java.util.Map; +import java.util.Random; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -import org.apache.hadoop.io.*; -import org.apache.hadoop.fs.*; -import org.apache.hadoop.conf.*; +import org.apache.hadoop.conf.Configuration; +import org.apache.hadoop.fs.FileStatus; +import org.apache.hadoop.fs.FileSystem; +import org.apache.hadoop.fs.Path; +import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; -import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; -import org.apache.hadoop.util.*; +import org.apache.hadoop.util.StringUtils; +import org.apache.hadoop.util.Tool; +import org.apache.hadoop.util.ToolRunner; import org.apache.nutch.metadata.Nutch; import org.apache.nutch.util.FSUtils; import org.apache.nutch.util.HadoopFSUtil; @@ -330,7 +338,7 @@ public int run(String[] args) throws Exception { } else if(args.containsKey(Nutch.ARG_SEGMENTS)) { Object segments = args.get(Nutch.ARG_SEGMENTS); - ArrayList<String> segmentList = new ArrayList<String>(); + ArrayList<String> segmentList = new ArrayList<>(); if(segments instanceof ArrayList) { segmentList = (ArrayList<String>)segments; } @@ -343,8 +351,8 @@ else if(segments instanceof Path){ } } else { - String segment_dir = crawlId+"/segments"; - File dir = new File(segment_dir); + String segmentDir = crawlId+"/segments"; + File dir = new File(segmentDir); File[] segmentsList = dir.listFiles(); Arrays.sort(segmentsList, (f1, f2) -> { if(f1.lastModified()>f2.lastModified()) diff --git a/src/java/org/apache/nutch/crawl/CrawlDbFilter.java b/src/java/org/apache/nutch/crawl/CrawlDbFilter.java index f143c7d73..56bc48260 100644 --- a/src/java/org/apache/nutch/crawl/CrawlDbFilter.java +++ b/src/java/org/apache/nutch/crawl/CrawlDbFilter.java @@ -23,7 +23,6 @@ import org.slf4j.Logger; import org.slf4j.LoggerFactory; import org.apache.hadoop.io.Text; -import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.conf.Configuration; import org.apache.nutch.net.URLFilters; diff --git a/src/java/org/apache/nutch/crawl/CrawlDbMerger.java b/src/java/org/apache/nutch/crawl/CrawlDbMerger.java index f8233aec9..35eca6069 100644 --- a/src/java/org/apache/nutch/crawl/CrawlDbMerger.java +++ b/src/java/org/apache/nutch/crawl/CrawlDbMerger.java @@ -20,27 +20,28 @@ import java.io.IOException; import java.lang.invoke.MethodHandles; import java.text.SimpleDateFormat; -import java.util.*; +import java.util.ArrayList; import java.util.Map.Entry; +import java.util.Random; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; import org.apache.hadoop.conf.Configuration; +import org.apache.hadoop.conf.Configured; import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.fs.Path; import org.apache.hadoop.io.Text; import org.apache.hadoop.io.Writable; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Reducer; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; -import org.apache.hadoop.util.*; -import org.apache.hadoop.conf.*; +import org.apache.hadoop.util.StringUtils; +import org.apache.hadoop.util.Tool; +import org.apache.hadoop.util.ToolRunner; import org.apache.nutch.util.LockUtil; import org.apache.nutch.util.NutchConfiguration; import org.apache.nutch.util.NutchJob; @@ -209,10 +210,10 @@ public int run(String[] args) throws Exception { boolean filter = false; boolean normalize = false; for (int i = 1; i < args.length; i++) { - if (args[i].equals("-filter")) { + if ("-filter".equals(args[i])) { filter = true; continue; - } else if (args[i].equals("-normalize")) { + } else if ("-normalize".equals(args[i])) { normalize = true; continue; } diff --git a/src/java/org/apache/nutch/crawl/CrawlDbReader.java b/src/java/org/apache/nutch/crawl/CrawlDbReader.java index db6a7d12e..165d4561c 100644 --- a/src/java/org/apache/nutch/crawl/CrawlDbReader.java +++ b/src/java/org/apache/nutch/crawl/CrawlDbReader.java @@ -26,7 +26,6 @@ import java.util.Date; import java.util.HashMap; -import java.util.Iterator; import java.util.Map; import java.util.Map.Entry; import java.util.Random; @@ -34,7 +33,6 @@ import java.util.regex.Pattern; import java.util.TreeMap; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; @@ -66,7 +64,6 @@ import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.RecordWriter; import org.apache.hadoop.mapreduce.TaskAttemptContext; -import org.apache.hadoop.util.Progressable; import org.apache.hadoop.util.Tool; import org.apache.hadoop.util.ToolRunner; import org.apache.hadoop.util.StringUtils; diff --git a/src/java/org/apache/nutch/crawl/CrawlDbReducer.java b/src/java/org/apache/nutch/crawl/CrawlDbReducer.java index b5851adbe..eb2729b85 100644 --- a/src/java/org/apache/nutch/crawl/CrawlDbReducer.java +++ b/src/java/org/apache/nutch/crawl/CrawlDbReducer.java @@ -19,21 +19,17 @@ import java.lang.invoke.MethodHandles; import java.util.ArrayList; -import java.util.Iterator; import java.util.List; import java.util.Map.Entry; -import java.io.IOException; -// Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -import org.apache.hadoop.io.*; -import org.apache.hadoop.mapreduce.Job; -import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; +import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Reducer; -import org.apache.hadoop.conf.*; +import org.apache.hadoop.conf.Configuration; +import org.apache.hadoop.conf.Configured; +import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; import org.apache.nutch.util.NutchConfiguration; import org.apache.nutch.util.NutchJob; import org.apache.nutch.util.NutchTool; @@ -41,7 +37,6 @@ import org.apache.nutch.util.TimingUtil; import org.apache.hadoop.util.Tool; import org.apache.hadoop.util.ToolRunner; -import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; import org.apache.hadoop.util.StringUtils; /* Reduce a CrawlDb by merging all CrawlDatum entries

org.apache.hadoop.io.Text; +import org.apache.hadoop.io.Writable; import org.apache.hadoop.util.PriorityQueue; import org.apache.nutch.metadata.Nutch; import org.apache.nutch.scoring.ScoringFilterException; diff --git a/src/java/org/apache/nutch/crawl/DeduplicationJob.java b/src/java/org/apache/nutch/crawl/DeduplicationJob.java index 0d03fa687..f2283ee9c 100644 --- a/src/java/org/apache/nutch/crawl/DeduplicationJob.java +++ b/src/java/org/apache/nutch/crawl/DeduplicationJob.java @@ -22,11 +22,8 @@ import java.net.URLDecoder; import java.text.SimpleDateFormat; import java.util.HashMap; -import java.util.Iterator; import java.util.Map; import java.util.Random; -import java.util.Arrays; - import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.fs.Path; import org.apache.hadoop.io.BytesWritable; @@ -39,7 +36,6 @@ import org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.CounterGroup; import org.apache.hadoop.mapreduce.Counter; import org.apache.hadoop.conf.Configuration; diff --git a/src/java/org/apache/nutch/crawl/Generator.java b/src/java/org/apache/nutch/crawl/Generator.java index 852cd3f7d..c972a13c0 100644 --- a/src/java/org/apache/nutch/crawl/Generator.java +++ b/src/java/org/apache/nutch/crawl/Generator.java @@ -17,20 +17,24 @@ package org.apache.nutch.crawl; -import java.io.*; +import java.io.DataInput; +import java.io.DataOutput; +import java.io.IOException; import java.lang.invoke.MethodHandles; -import java.net.*; -import java.util.*; -import java.text.*; +import java.net.URL; +import java.text.SimpleDateFormat; +import java.util.ArrayList; +import java.util.Date; +import java.util.HashMap; +import java.util.List; +import java.util.Map; +import java.util.Random; // rLogging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; import org.apache.commons.jexl2.Expression; import org.apache.commons.jexl2.JexlContext; import org.apache.commons.jexl2.MapContext; -import org.apache.hadoop.io.*; -import org.apache.hadoop.conf.*; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Reducer; @@ -38,13 +42,24 @@ import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; +import org.apache.hadoop.util.StringUtils; +import org.apache.hadoop.util.Tool; +import org.apache.hadoop.util.ToolRunner; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.hadoop.mapreduce.Partitioner; import org.apache.hadoop.mapreduce.lib.output.MultipleOutputs; -import org.apache.hadoop.util.*; +import org.apache.hadoop.io.*; +import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.fs.FileStatus; import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.fs.Path; +import org.apache.hadoop.io.FloatWritable; +import org.apache.hadoop.io.IntWritable; +import org.apache.hadoop.io.LongWritable; +import org.apache.hadoop.io.MapFile; +import org.apache.hadoop.io.Text; +import org.apache.hadoop.io.Writable; +import org.apache.hadoop.io.WritableComparable; +import org.apache.hadoop.io.WritableComparator; import org.apache.nutch.hostdb.HostDatum; import org.apache.nutch.metadata.Nutch; import org.apache.nutch.net.URLFilterException; diff --git a/src/java/org/apache/nutch/crawl/Inlink.java b/src/java/org/apache/nutch/crawl/Inlink.java index 67df357cb..631f8bf17 100644 --- a/src/java/org/apache/nutch/crawl/Inlink.java +++ b/src/java/org/apache/nutch/crawl/Inlink.java @@ -17,8 +17,12 @@ package org.apache.nutch.crawl; -import java.io.*; -import org.apache.hadoop.io.*; +import java.io.DataInput; +import java.io.DataOutput; +import java.io.IOException; ++import org.apache.hadoop.io.Text; +import org.apache.hadoop.io.Writable; /* An incoming link to a page. */ public class Inlink implements Writable { diff --git a/src/java/org/apache/nutch/crawl/Inlinks.java b/src/java/org/apache/nutch/crawl/Inlinks.java index 9ce7a8522..42dd9db11 100644 --- a/src/java/org/apache/nutch/crawl/Inlinks.java +++ b/src/java/org/apache/nutch/crawl/Inlinks.java @@ -17,11 +17,18 @@ package org.apache.nutch.crawl; -import java.io.*; -import java.net.*; -import java.util.*; - -import org.apache.hadoop.io.*; +import java.io.DataInput; +import java.io.DataOutput; +import java.io.IOException; +import java.net.MalformedURLException; +import java.net.URL; +import java.util.ArrayList; +import java.util.HashMap; +import java.util.HashSet; +import java.util.Iterator; +import java.util.Set; ++import org.apache.hadoop.io.Writable; /** A list of {@link Inlink}s. */ public class Inlinks implements Writable { @@ -64,7 +71,7 @@ public void write(DataOutput out) throws IOException { } public String toString() { - StringBuffer buffer = new StringBuffer(); + StringBuilder buffer = new StringBuilder(); buffer.append("Inlinks:\n"); Iterator<Inlink> it = inlinks.iterator(); while (it.hasNext()) { diff --git a/src/java/org/apache/nutch/crawl/LinkDbFilter.java b/src/java/org/apache/nutch/crawl/LinkDbFilter.java index 389535a2f..757cfde12 100644 --- a/src/java/org/apache/nutch/crawl/LinkDbFilter.java +++ b/src/java/org/apache/nutch/crawl/LinkDbFilter.java @@ -25,9 +25,7 @@ import org.slf4j.LoggerFactory; import org.apache.hadoop.io.Text; import org.apache.hadoop.conf.Configuration; -import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.nutch.net.URLFilters; import org.apache.nutch.net.URLNormalizers; diff --git a/src/java/org/apache/nutch/crawl/LinkDbMerger.java b/src/java/org/apache/nutch/crawl/LinkDbMerger.java index 4191db709..c8e3943b2 100644 --- a/src/java/org/apache/nutch/crawl/LinkDbMerger.java +++ b/src/java/org/apache/nutch/crawl/LinkDbMerger.java @@ -32,7 +32,6 @@ import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Reducer; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import

org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; diff --git a/src/java/org/apache/nutch/crawl/LinkDbReader.java b/src/java/org/apache/nutch/crawl/LinkDbReader.java index 717973a16..bf537b73a 100644 --- a/src/java/org/apache/nutch/crawl/LinkDbReader.java +++ b/src/java/org/apache/nutch/crawl/LinkDbReader.java @@ -28,11 +28,14 @@ import org.slf4j.LoggerFactory; import org.apache.hadoop.conf.Configured; -import org.apache.hadoop.io.*; -import org.apache.hadoop.fs.*; +import org.apache.hadoop.fs.FileSystem; +import org.apache.hadoop.fs.Path; +import org.apache.hadoop.io.MapFile; +import org.apache.hadoop.io.Text; +import org.apache.hadoop.io.Writable; +import org.apache.hadoop.io.WritableComparable; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; @@ -40,7 +43,9 @@ import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat; import org.apache.hadoop.mapreduce.Partitioner; import org.apache.hadoop.mapreduce.lib.partition.HashPartitioner; -import org.apache.hadoop.util.*; +import org.apache.hadoop.util.StringUtils; +import org.apache.hadoop.util.Tool; +import org.apache.hadoop.util.ToolRunner; import org.apache.hadoop.conf.Configuration; import org.apache.nutch.util.NutchConfiguration; @@ -63,7 +68,7 @@ private MapFile.Reader[] readers; public LinkDbReader() { - + //default constructor } public LinkDbReader(Configuration conf, Path directory) throws Exception { @@ -167,8 +172,8 @@ public void processDumpJob(String linkdb, String output, String regex) } long end = System.currentTimeMillis(); - LOG.info("LinkDb dump: finished at " + sdf.format(end) + ", elapsed: " + - TimingUtil.elapsedTime(start, end)); + LOG.info("LinkDb dump: finished at {}, elapsed: {}", + sdf.format(end), TimingUtil.elapsedTime(start, end)); } public static void main(String[] args) throws Exception { diff --git a/src/java/org/apache/nutch/crawl/SignatureFactory.java b/src/java/org/apache/nutch/crawl/SignatureFactory.java index 1966ca229..6832ffc61 100644 --- a/src/java/org/apache/nutch/crawl/SignatureFactory.java +++ b/src/java/org/apache/nutch/crawl/SignatureFactory.java @@ -17,7 +17,6 @@ package org.apache.nutch.crawl; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/java/org/apache/nutch/crawl/URLPartitioner.java b/src/java/org/apache/nutch/crawl/URLPartitioner.java index dfb31b310..cc508962e 100644 --- a/src/java/org/apache/nutch/crawl/URLPartitioner.java +++ b/src/java/org/apache/nutch/crawl/URLPartitioner.java @@ -25,9 +25,10 @@ import org.slf4j.Logger; import org.slf4j.LoggerFactory; -import org.apache.hadoop.io.*; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.conf.Configuration; +import org.apache.hadoop.io.Text; +import org.apache.hadoop.io.Writable; import org.apache.nutch.net.URLNormalizers; import org.apache.nutch.util.URLUtil; import org.apache.hadoop.mapreduce.Partitioner; diff --git a/src/java/org/apache/nutch/fetcher/FetchNodeDb.java b/src/java/org/apache/nutch/fetcher/FetchNodeDb.java index 1a073ab57..5fdde70e2 100644 --- a/src/java/org/apache/nutch/fetcher/FetchNodeDb.java +++ b/src/java/org/apache/nutch/fetcher/FetchNodeDb.java @@ -19,7 +19,6 @@ import java.util.Map; import java.util.concurrent.ConcurrentHashMap; - public class FetchNodeDb { private Map<Integer, FetchNode> fetchNodeDbMap; diff --git a/src/java/org/apache/nutch/fetcher/Fetcher.java b/src/java/org/apache/nutch/fetcher/Fetcher.java index 4320f5532..34fb136d2 100644 --- a/src/java/org/apache/nutch/fetcher/Fetcher.java +++ b/src/java/org/apache/nutch/fetcher/Fetcher.java @@ -20,26 +20,29 @@ import java.io.IOException; import java.lang.invoke.MethodHandles; import java.text.SimpleDateFormat; -import java.util.*; +import java.util.ArrayList; +import java.util.Arrays; +import java.util.HashMap; +import java.util.Iterator; +import java.util.LinkedList; +import java.util.List; +import java.util.Map; import java.util.concurrent.atomic.AtomicInteger; import java.util.concurrent.atomic.AtomicLong; import org.slf4j.Logger; import org.slf4j.LoggerFactory; -import org.apache.hadoop.io.*; -import org.apache.hadoop.fs.*; +import org.apache.hadoop.conf.Configurable; import org.apache.hadoop.conf.Configuration; +import org.apache.hadoop.fs.FileStatus; +import org.apache.hadoop.fs.Path; +import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.JobContext; import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.Reducer; -import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; -import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; -import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; import org.apache.hadoop.mapreduce.RecordReader; import org.apache.hadoop.mapreduce.InputSplit; -import org.apache.hadoop.mapred.FileSplit; import org.apache.hadoop.util.StringUtils; @@ -48,8 +51,10 @@ import org.apache.nutch.crawl.NutchWritable; import org.apache.nutch.metadata.Nutch; import org.apache.nutch.protocol.*; -import org.apache.nutch.util.*; +import org.apache.nutch.util.NutchConfiguration; +import org.apache.nutch.util.NutchJob; +import org.apache.nutch.util.NutchTool; import org.apache.nutch.util.TimingUtil; /** * A queue-based fetcher. @@ -206,7 +211,6 @@ public void run(Context innerContext) throws IOException { feeder = new QueueFeeder(innerContext, fetchQueues, threadCount * queueDepthMuliplier); - // feeder.setPriority((Thread.MAX_PRIORITY + Thread.NORM_PRIORITY) / 2); // the value of the time limit is either -1 or the time where it should // finish @@ -576,8 +580,8 @@ else if(seg instanceof ArrayList) { } } else { - String segment_dir = crawlId+"/segments"; - File segmentsDir = new File(segment_dir); +

String segmentDir = crawlId+"/segments"; + File segmentsDir = new File(segmentDir); File[] segmentsList = segmentsDir.listFiles(); Arrays.sort(segmentsList, (f1, f2) -> { if(f1.lastModified()>f2.lastModified()) diff --git a/src/java/org/apache/nutch/fetcher/FetcherOutputFormat.java b/src/java/org/apache/nutch/fetcher/FetcherOutputFormat.java index 9117e4bb6..11c09ae1d 100644 --- a/src/java/org/apache/nutch/fetcher/FetcherOutputFormat.java +++ b/src/java/org/apache/nutch/fetcher/FetcherOutputFormat.java @@ -33,15 +33,10 @@ import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapred.InvalidJobConfException; -import org.apache.hadoop.mapreduce.OutputFormat; import org.apache.hadoop.mapreduce.RecordWriter; -import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; import org.apache.hadoop.mapreduce.TaskAttemptContext; import org.apache.hadoop.mapreduce.JobContext; -import org.apache.hadoop.mapreduce.InputSplit; -import org.apache.hadoop.mapred.FileSplit; import org.apache.hadoop.util.Progressable; import org.apache.nutch.parse.Parse; import org.apache.nutch.parse.ParseOutputFormat; import org.apache.nutch.protocol.Content; diff --git a/src/java/org/apache/nutch/fetcher/FetcherThread.java b/src/java/org/apache/nutch/fetcher/FetcherThread.java index d894c8b44..ad5cbba22 100644 --- a/src/java/org/apache/nutch/fetcher/FetcherThread.java +++ b/src/java/org/apache/nutch/fetcher/FetcherThread.java @@ -32,8 +32,6 @@ import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Mapper.Context; -//import org.apache.hadoop.mapred.OutputCollector; -//import org.apache.hadoop.mapred.Reporter; import org.apache.hadoop.util.StringUtils; import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.crawl.NutchWritable; diff --git a/src/java/org/apache/nutch/fetcher/FetcherThreadEvent.java b/src/java/org/apache/nutch/fetcher/FetcherThreadEvent.java index 26dc9466c..6c175c83a 100644 --- a/src/java/org/apache/nutch/fetcher/FetcherThreadEvent.java +++ b/src/java/org/apache/nutch/fetcher/FetcherThreadEvent.java @@ -22,7 +22,6 @@ import java.util.HashMap; import java.util.Map; -import org.apache.nutch.metadata.Nutch; import org.apache.nutch.parse.Outlink; /** diff --git a/src/java/org/apache/nutch/fetcher/QueueFeeder.java b/src/java/org/apache/nutch/fetcher/QueueFeeder.java index 738fceeb1..de02c4847 100644 --- a/src/java/org/apache/nutch/fetcher/QueueFeeder.java +++ b/src/java/org/apache/nutch/fetcher/QueueFeeder.java @@ -20,7 +20,6 @@ import java.lang.invoke.MethodHandles; import org.apache.hadoop.io.Text; -import org.apache.hadoop.mapreduce.RecordReader; import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.nutch.crawl.CrawlDatum; import org.slf4j.Logger; diff --git a/src/java/org/apache/nutch/hostdb/HostDatum.java b/src/java/org/apache/nutch/hostdb/HostDatum.java index 65de170f0..fe3b73e65 100644 --- a/src/java/org/apache/nutch/hostdb/HostDatum.java +++ b/src/java/org/apache/nutch/hostdb/HostDatum.java @@ -24,8 +24,6 @@ import java.text.SimpleDateFormat; import org.apache.hadoop.io.MapWritable; -import org.apache.hadoop.io.IntWritable; -import org.apache.hadoop.io.LongWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.io.Writable; diff --git a/src/java/org/apache/nutch/hostdb/ReadHostDb.java b/src/java/org/apache/nutch/hostdb/ReadHostDb.java index c28ff6f7f..e53e0c3ed 100644 --- a/src/java/org/apache/nutch/hostdb/ReadHostDb.java +++ b/src/java/org/apache/nutch/hostdb/ReadHostDb.java @@ -26,8 +26,6 @@ import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.conf.Configured; import org.apache.hadoop.fs.Path; -import org.apache.hadoop.fs.FileStatus; -import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.io.FloatWritable; import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.SequenceFile; @@ -36,7 +34,6 @@ import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; -import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat; import org.apache.hadoop.mapreduce.Mapper; @@ -45,9 +42,7 @@ import org.apache.hadoop.util.Tool; import org.apache.hadoop.util.ToolRunner; import org.apache.nutch.util.NutchConfiguration; -import org.apache.nutch.util.StringUtil; import org.apache.nutch.util.TimingUtil; -import org.apache.nutch.util.URLUtil; import org.apache.nutch.util.SegmentReaderUtil; import org.apache.commons.jexl2.JexlContext; diff --git a/src/java/org/apache/nutch/hostdb/UpdateHostDb.java b/src/java/org/apache/nutch/hostdb/UpdateHostDb.java index c46f78879..578c74af0 100644 --- a/src/java/org/apache/nutch/hostdb/UpdateHostDb.java +++ b/src/java/org/apache/nutch/hostdb/UpdateHostDb.java @@ -23,14 +23,9 @@ import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.conf.Configured; import org.apache.hadoop.fs.Path; -import org.apache.hadoop.fs.FileStatus; import org.apache.hadoop.fs.FileSystem; -import org.apache.hadoop.fs.Path; import org.apache.hadoop.io.Text; -import org.apache.hadoop.io.Writable; -import org.apache.hadoop.io.WritableUtils; import org.apache.hadoop.mapreduce.Job; -import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; @@ -39,7 +34,6 @@ import java.lang.invoke.MethodHandles;

org.apache.hadoop.util.StringUtils; import org.apache.hadoop.util.Tool; import org.apache.hadoop.util.ToolRunner; -import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.crawl.CrawlDb; import org.apache.nutch.crawl.NutchWritable; import org.apache.nutch.util.FSUtils; diff --git a/src/java/org/apache/nutch/hostdb/UpdateHostDbMapper.java b/src/java/org/apache/nutch/hostdb/UpdateHostDbMapper.java index 8f0be7641..1c3c8c323 100644 --- a/src/java/org/apache/nutch/hostdb/UpdateHostDbMapper.java +++ b/src/java/org/apache/nutch/hostdb/UpdateHostDbMapper.java @@ -19,17 +19,13 @@ import java.io.IOException; import java.lang.invoke.MethodHandles; - import org.apache.hadoop.io.FloatWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.io.Writable; -import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.conf.Configuration; import org.apache.nutch.crawl.CrawlDatum; -import org.apache.nutch.crawl.CrawlDb; import org.apache.nutch.crawl.NutchWritable; import org.apache.nutch.metadata.Nutch; import org.apache.nutch.net.URLFilters; diff --git a/src/java/org/apache/nutch/hostdb/UpdateHostDbReducer.java b/src/java/org/apache/nutch/hostdb/UpdateHostDbReducer.java index 4e829b51d..34a51037e 100644 --- a/src/java/org/apache/nutch/hostdb/UpdateHostDbReducer.java +++ b/src/java/org/apache/nutch/hostdb/UpdateHostDbReducer.java @@ -19,7 +19,6 @@ import java.io.IOException; import java.lang.invoke.MethodHandles; import java.util.Date; -import java.util.Iterator; import java.util.concurrent.BlockingQueue; import java.util.concurrent.SynchronousQueue; import java.util.concurrent.ThreadPoolExecutor; @@ -32,9 +31,7 @@ import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.io.Writable; -import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Reducer; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.util.StringUtils; import org.apache.nutch.crawl.CrawlDatum; diff --git a/src/java/org/apache/nutch/indexer/CleaningJob.java b/src/java/org/apache/nutch/indexer/CleaningJob.java index 720862e9a..a8ac64044 100644 --- a/src/java/org/apache/nutch/indexer/CleaningJob.java +++ b/src/java/org/apache/nutch/indexer/CleaningJob.java @@ -19,8 +19,6 @@ import java.io.IOException; import java.lang.invoke.MethodHandles; import java.text.SimpleDateFormat; -import java.util.Iterator; - import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.fs.Path; import org.apache.hadoop.io.ByteWritable; @@ -29,7 +27,6 @@ import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Reducer; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.hadoop.mapreduce.lib.output.NullOutputFormat; import org.apache.hadoop.util.StringUtils; diff --git a/src/java/org/apache/nutch/indexer/IndexWriter.java b/src/java/org/apache/nutch/indexer/IndexWriter.java index 31056ef3b..7cc2d1569 100644 --- a/src/java/org/apache/nutch/indexer/IndexWriter.java +++ b/src/java/org/apache/nutch/indexer/IndexWriter.java @@ -20,7 +20,6 @@ import org.apache.hadoop.conf.Configurable; import org.apache.hadoop.conf.Configuration; -import org.apache.hadoop.mapreduce.Job; import org.apache.nutch.indexer.NutchDocument; import org.apache.nutch.plugin.Pluggable; diff --git a/src/java/org/apache/nutch/indexer/IndexWriters.java b/src/java/org/apache/nutch/indexer/IndexWriters.java index 6ecf8c942..63ddaffa5 100644 --- a/src/java/org/apache/nutch/indexer/IndexWriters.java +++ b/src/java/org/apache/nutch/indexer/IndexWriters.java @@ -21,7 +21,6 @@ import java.util.HashMap; import org.apache.hadoop.conf.Configuration; -import org.apache.hadoop.mapreduce.Job; import org.apache.nutch.indexer.NutchDocument; import org.apache.nutch.plugin.Extension; import org.apache.nutch.plugin.ExtensionPoint; diff --git a/src/java/org/apache/nutch/indexer/IndexerMapReduce.java b/src/java/org/apache/nutch/indexer/IndexerMapReduce.java index 0b6faeb61..bc1c82a16 100644 --- a/src/java/org/apache/nutch/indexer/IndexerMapReduce.java +++ b/src/java/org/apache/nutch/indexer/IndexerMapReduce.java @@ -19,22 +19,17 @@ import java.io.IOException; import java.lang.invoke.MethodHandles; import java.util.Collection; -import java.util.Iterator; - import org.slf4j.Logger; import org.slf4j.LoggerFactory; import org.apache.commons.codec.binary.Base64; -import org.apache.commons.codec.binary.StringUtils; import org.apache.hadoop.conf.Configured; import org.apache.hadoop.conf.Configuration; -import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.fs.Path; import org.apache.hadoop.io.Text; import org.apache.hadoop.io.Writable; import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.nutch.crawl.CrawlDatum; diff --git a/src/java/org/apache/nutch/indexer/IndexerOutputFormat.java b/src/java/org/apache/nutch/indexer/IndexerOutputFormat.java index c220efb09..54b98dfce 100644 --- a/src/java/org/apache/nutch/indexer/IndexerOutputFormat.java +++ b/src/java/org/apache/nutch/indexer/IndexerOutputFormat.java @@ -20,7 +20,6 @@ import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.Job; -import org.apache.hadoop.mapreduce.OutputFormat; import org.apache.hadoop.mapreduce.RecordWriter; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.mapreduce.TaskAttemptContext; diff --git

a/src/java/org/apache/nutch/indexer/IndexingFilter.java b/src/java/org/apache/nutch/indexer/IndexingFilter.java index f22a0e51d..b34b9b7f4 100644 --- a/src/java/org/apache/nutch/indexer/IndexingFilter.java +++ b/src/java/org/apache/nutch/indexer/IndexingFilter.java @@ -17,11 +17,9 @@ package org.apache.nutch.indexer; -// Hadoop imports import org.apache.hadoop.conf.Configurable; import org.apache.hadoop.io.Text; -// Nutch imports import org.apache.nutch.parse.Parse; import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.crawl.Inlinks; diff --git a/src/java/org/apache/nutch/indexer/IndexingFilters.java b/src/java/org/apache/nutch/indexer/IndexingFilters.java index 439658b4f..ca603d4f0 100644 --- a/src/java/org/apache/nutch/indexer/IndexingFilters.java +++ b/src/java/org/apache/nutch/indexer/IndexingFilters.java @@ -17,7 +17,6 @@ package org.apache.nutch.indexer; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/java/org/apache/nutch/indexer/IndexingFiltersChecker.java b/src/java/org/apache/nutch/indexer/IndexingFiltersChecker.java index 9662bca0b..e83bf3e0a 100644 --- a/src/java/org/apache/nutch/indexer/IndexingFiltersChecker.java +++ b/src/java/org/apache/nutch/indexer/IndexingFiltersChecker.java @@ -24,7 +24,6 @@ import java.util.Map; import org.apache.hadoop.io.Text; -import org.apache.hadoop.util.Tool; import org.apache.hadoop.util.ToolRunner; import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.crawl.Inlinks; diff --git a/src/java/org/apache/nutch/indexer/IndexingJob.java b/src/java/org/apache/nutch/indexer/IndexingJob.java index 245132bee..e4997cb89 100644 --- a/src/java/org/apache/nutch/indexer/IndexingJob.java +++ b/src/java/org/apache/nutch/indexer/IndexingJob.java @@ -22,7 +22,6 @@ import java.text.SimpleDateFormat; import java.util.ArrayList; import java.util.Arrays; -import java.util.Comparator; import java.util.HashMap; import java.util.List; import java.util.Locale; diff --git a/src/java/org/apache/nutch/indexer/NutchField.java b/src/java/org/apache/nutch/indexer/NutchField.java index d70919323..d6b2b3b2c 100644 --- a/src/java/org/apache/nutch/indexer/NutchField.java +++ b/src/java/org/apache/nutch/indexer/NutchField.java @@ -25,7 +25,8 @@ import java.util.Date; import java.util.List; -import org.apache.hadoop.io.*; +import org.apache.hadoop.io.Text; +import org.apache.hadoop.io.Writable; /** * This class represents a multi-valued field with a weight. Values are @@ -36,6 +37,7 @@ private List<Object> values = new ArrayList<>(); public NutchField() { + //default constructor } public NutchField(Object value) { @@ -89,17 +91,17 @@ public void readFields(DataInput in) throws IOException { for (int i = 0; i < count; i++) { String type = Text.readString(in); - if (type.equals("java.lang.String")) { + if ("java.lang.String".equals(type)) { values.add(Text.readString(in)); - } else if (type.equals("java.lang.Boolean")) { + } else if ("java.lang.Boolean".equals(type)) { values.add(in.readBoolean()); - } else if (type.equals("java.lang.Integer")) { + } else if ("java.lang.Integer".equals(type)) { values.add(in.readInt()); - } else if (type.equals("java.lang.Float")) { + } else if ("java.lang.Float".equals(type)) { values.add(in.readFloat()); - } else if (type.equals("java.lang.Long")) { + } else if ("java.lang.Long".equals(type)) { values.add(in.readLong()); - } else if (type.equals("java.util.Date")) { + } else if ("java.util.Date".equals(type)) { values.add(new Date(in.readLong())); } } @@ -130,6 +132,7 @@ public void write(DataOutput out) throws IOException { } } + @Override public String toString() { return values.toString(); } diff --git a/src/java/org/apache/nutch/metadata/CreativeCommons.java b/src/java/org/apache/nutch/metadata/CreativeCommons.java index f9c425bc0..37a36a948 100644 --- a/src/java/org/apache/nutch/metadata/CreativeCommons.java +++ b/src/java/org/apache/nutch/metadata/CreativeCommons.java @@ -26,10 +26,10 @@ */ public interface CreativeCommons { - public final static String LICENSE_URL = "License-Url"; + public static final String LICENSE_URL = "License-Url"; - public final static String LICENSE_LOCATION = "License-Location"; + public static final String LICENSE_LOCATION = "License-Location"; - public final static String WORK_TYPE = "Work-Type"; + public static final String WORK_TYPE = "Work-Type"; } diff --git a/src/java/org/apache/nutch/metadata/HttpHeaders.java b/src/java/org/apache/nutch/metadata/HttpHeaders.java index 78b87972d..71a66f66c 100644 --- a/src/java/org/apache/nutch/metadata/HttpHeaders.java +++ b/src/java/org/apache/nutch/metadata/HttpHeaders.java @@ -26,26 +26,26 @@ */ public interface HttpHeaders { - public final static String TRANSFER_ENCODING = "Transfer-Encoding"; + public static final String TRANSFER_ENCODING = "Transfer-Encoding"; - public final static String CONTENT_ENCODING = "Content-Encoding"; + public static final String CONTENT_ENCODING = "Content-Encoding"; - public final static String CONTENT_LANGUAGE = "Content-Language"; + public static final String CONTENT_LANGUAGE = "Content-Language"; - public final static String CONTENT_LENGTH = "Content-Length"; + public static final String CONTENT_LENGTH = "Content-Length"; - public final static String CONTENT_LOCATION = "Content-Location"; + public static final String CONTENT_LOCATION = "Content-Location"; - public final static String CONTENT_DISPOSITION = "Content-Disposition"; + public static final String CONTENT_DISPOSITION = "Content-Disposition"; - public final static String CONTENT_MD5 = "Content-MD5"; + public static final String CONTENT_MD5 = "Content-MD5"; - public final static String CONTENT_TYPE = "Content-Type"; + public static final String CONTENT_TYPE = "Content-Type"; - public final static Text WRITABLE_CONTENT_TYPE = new Text(CONTENT_TYPE); + public static final Text WRITABLE_CONTENT_TYPE = new Text(CONTENT_TYPE); - public final static String LAST_MODIFIED = "Last-Modified"; + public static final String LAST_MODIFIED = "Last-Modified"; - public final static String LOCATION = "Location"; + public static final String LOCATION = "Location"; } diff --git a/src/java/org/apache/nutch/net/URLExemptionFilter.java b/src/java/org/apache/nutch/net/URLExemptionFilter.java index 3b416d2c6..03b3f61ea 100644 --- a/src/java/org/apache/nutch/net/URLExemptionFilter.java +++ b/src/java/org/apache/nutch/net/URLExemptionFilter.java @@ -17,9 +17,8 @@ package org.apache.nutch.net; -// Hadoop import org.apache.hadoop.conf.Configurable; -// Nutch +import org.apache.nutch.plugin.Pluggable; /**

diff --git a/src/java/org/apache/nutch/net/URLFilter.java b/src/java/org/apache/nutch/net/URLFilter.java index 01efbcdfe..7fabc5f65 100644 --- a/src/java/org/apache/nutch/net/URLFilter.java +++ b/src/java/org/apache/nutch/net/URLFilter.java @@ -17,10 +17,8 @@ package org.apache.nutch.net; -// Hadoop imports import org.apache.hadoop.conf.Configurable; -// Nutch imports import org.apache.nutch.plugin.Pluggable; /** diff --git a/src/java/org/apache/nutch/net/URLFilterChecker.java b/src/java/org/apache/nutch/net/URLFilterChecker.java index ceca301b0..52e557fdd 100644 --- a/src/java/org/apache/nutch/net/URLFilterChecker.java +++ b/src/java/org/apache/nutch/net/URLFilterChecker.java @@ -17,18 +17,11 @@ package org.apache.nutch.net; -import org.apache.nutch.plugin.Extension; -import org.apache.nutch.plugin.ExtensionPoint; -import org.apache.nutch.plugin.PluginRepository; - import org.apache.hadoop.util.ToolRunner; import org.apache.nutch.util.AbstractChecker; import org.apache.nutch.util.NutchConfiguration; -import java.io.BufferedReader; -import java.io.InputStreamReader; - /** * Checks one given filter or all filters. * diff --git a/src/java/org/apache/nutch/net/URLNormalizerChecker.java b/src/java/org/apache/nutch/net/URLNormalizerChecker.java index 64fae5896..bd3ca5eb5 100644 --- a/src/java/org/apache/nutch/net/URLNormalizerChecker.java +++ b/src/java/org/apache/nutch/net/URLNormalizerChecker.java @@ -17,18 +17,11 @@ package org.apache.nutch.net; -import org.apache.nutch.plugin.Extension; -import org.apache.nutch.plugin.ExtensionPoint; -import org.apache.nutch.plugin.PluginRepository; - import org.apache.hadoop.util.ToolRunner; import org.apache.nutch.util.AbstractChecker; import org.apache.nutch.util.NutchConfiguration; -import java.io.BufferedReader; -import java.io.InputStreamReader; - /** * Checks one given normalizer or all normalizers. */ diff --git a/src/java/org/apache/nutch/net/protocols/Response.java b/src/java/org/apache/nutch/net/protocols/Response.java index efff14b4c..c9139bd6c 100644 --- a/src/java/org/apache/nutch/net/protocols/Response.java +++ b/src/java/org/apache/nutch/net/protocols/Response.java @@ -16,10 +16,8 @@ */ package org.apache.nutch.net.protocols; -// JDK imports import java.net.URL; -// Nutch imports import org.apache.nutch.metadata.HttpHeaders; import org.apache.nutch.metadata.Metadata; diff --git a/src/java/org/apache/nutch/parse/HtmlParseFilter.java b/src/java/org/apache/nutch/parse/HtmlParseFilter.java index 55b51aca6..2238949ef 100644 --- a/src/java/org/apache/nutch/parse/HtmlParseFilter.java +++ b/src/java/org/apache/nutch/parse/HtmlParseFilter.java @@ -17,13 +17,10 @@ package org.apache.nutch.parse; -// JDK imports import org.w3c.dom.DocumentFragment; -// Hadoop imports import org.apache.hadoop.conf.Configurable; -// Nutch imports import org.apache.nutch.plugin.Pluggable; import org.apache.nutch.protocol.Content; diff --git a/src/java/org/apache/nutch/parse/ParseData.java b/src/java/org/apache/nutch/parse/ParseData.java index 028682ec6..f80a5bc73 100644 --- a/src/java/org/apache/nutch/parse/ParseData.java +++ b/src/java/org/apache/nutch/parse/ParseData.java @@ -17,16 +17,20 @@ package org.apache.nutch.parse; -import java.io.*; -import java.util.*; +import java.io.DataInput; +import java.io.DataOutput; +import java.io.IOException; +import java.util.Arrays; import org.apache.commons.cli.Options; -import org.apache.hadoop.io.*; import org.apache.hadoop.util.GenericOptionsParser; import org.apache.hadoop.conf.Configuration; -import org.apache.hadoop.fs.*; -import org.apache.hadoop.fs.FileSystem; - +import org.apache.hadoop.fs.Path; +import org.apache.hadoop.io.ArrayFile; +import org.apache.hadoop.io.Text; +import org.apache.hadoop.io.VersionMismatchException; +import org.apache.hadoop.io.VersionedWritable; import org.apache.nutch.metadata.Metadata; import org.apache.nutch.util.NutchConfiguration; @@ -38,7 +42,7 @@ public final class ParseData extends VersionedWritable { public static final String DIR_NAME = "parse_data"; - private final static byte VERSION = 5; + private static final byte VERSION = 5; private String title; private Outlink[] outlinks; diff --git a/src/java/org/apache/nutch/parse/ParseImpl.java b/src/java/org/apache/nutch/parse/ParseImpl.java index dc72769eb..77dbe7b4c 100644 --- a/src/java/org/apache/nutch/parse/ParseImpl.java +++ b/src/java/org/apache/nutch/parse/ParseImpl.java @@ -17,8 +17,11 @@ package org.apache.nutch.parse; -import java.io.*; -import org.apache.hadoop.io.*; +import java.io.DataInput; +import java.io.DataOutput; +import java.io.IOException; + +import org.apache.hadoop.io.Writable; /** * The result of parsing a page's raw content. diff --git a/src/java/org/apache/nutch/parse/ParseOutputFormat.java b/src/java/org/apache/nutch/parse/ParseOutputFormat.java index 2a454eb9b..d24f9ce4b 100644 --- a/src/java/org/apache/nutch/parse/ParseOutputFormat.java +++ b/src/java/org/apache/nutch/parse/ParseOutputFormat.java @@ -19,19 +19,20 @@ import java.text.NumberFormat; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; - -import org.apache.hadoop.io.*; +import org.apache.hadoop.io.MapFile; import org.apache.hadoop.io.MapFile.Writer.Option; +import org.apache.hadoop.io.MapWritable; import org.apache.hadoop.io.SequenceFile; import org.apache.hadoop.io.SequenceFile.CompressionType; import org.apache.hadoop.io.SequenceFile.Metadata; +import org.apache.hadoop.io.Text; import org.apache.hadoop.io.compress.DefaultCodec; import org.apache.hadoop.util.Progressable; -import org.apache.hadoop.fs.*; +import org.apache.hadoop.fs.FileSystem; +import org.apache.hadoop.fs.Path; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.OutputFormat; @@ -48,9 +49,11 @@ import org.apache.nutch.util.StringUtil; import org.apache.nutch.util.URLUtil; import org.apache.nutch.metadata.Nutch;

import org.apache.nutch.net.*; +import org.apache.nutch.net.URLExemptionFilters; +import org.apache.nutch.net.URLFilters; +import org.apache.nutch.net.URLNormalizers; -import java.io.*; +import java.io.IOException; import java.lang.invoke.MethodHandles; import java.net.MalformedURLException; import java.net.URL; @@ -58,8 +61,6 @@ import java.util.List; import java.util.Map.Entry; -import org.apache.hadoop.util.Progressable; - /* Parse content in a segment. */ public class ParseOutputFormat extends OutputFormat<Text, Parse> { private static final Logger LOG = LoggerFactory diff --git a/src/java/org/apache/nutch/parse/ParsePluginList.java b/src/java/org/apache/nutch/parse/ParsePluginList.java index a774d47d0..b4355a45c 100644 --- a/src/java/org/apache/nutch/parse/ParsePluginList.java +++ b/src/java/org/apache/nutch/parse/ParsePluginList.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.parse; -// JDK imports import java.util.Arrays; import java.util.HashMap; import java.util.List; diff --git a/src/java/org/apache/nutch/parse/ParsePluginsReader.java b/src/java/org/apache/nutch/parse/ParsePluginsReader.java index f0fb3a9e9..a3c5f84fa 100644 --- a/src/java/org/apache/nutch/parse/ParsePluginsReader.java +++ b/src/java/org/apache/nutch/parse/ParsePluginsReader.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.parse; -// JDK imports import java.io.InputStream; import java.lang.invoke.MethodHandles; import java.net.URL; @@ -33,14 +32,11 @@ import org.w3c.dom.NodeList; import org.xml.sax.InputSource; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Hadoop imports import org.apache.hadoop.conf.Configuration; -// Nutch imports import org.apache.nutch.util.NutchConfiguration; /** diff --git a/src/java/org/apache/nutch/parse/ParseSegment.java b/src/java/org/apache/nutch/parse/ParseSegment.java index af81df3a7..61aa99740 100644 --- a/src/java/org/apache/nutch/parse/ParseSegment.java +++ b/src/java/org/apache/nutch/parse/ParseSegment.java @@ -22,31 +22,41 @@ import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.crawl.SignatureFactory; import org.apache.nutch.segment.SegmentChecker; -import org.apache.hadoop.fs.FileSystem; -import org.apache.hadoop.io.*; +import org.apache.nutch.util.NutchConfiguration; +import org.apache.nutch.util.NutchJob; +import org.apache.nutch.util.NutchTool; +import org.apache.nutch.util.StringUtil; +import org.apache.nutch.util.TimingUtil; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Reducer; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; -import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; +import org.apache.hadoop.util.StringUtils; +import org.apache.hadoop.util.Tool; +import org.apache.hadoop.util.ToolRunner; import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; -import org.apache.hadoop.util.*; -import org.apache.hadoop.conf.*; import org.apache.nutch.metadata.Metadata; import org.apache.nutch.metadata.Nutch; import org.apache.nutch.net.protocols.Response; -import org.apache.nutch.protocol.*; +import org.apache.nutch.protocol.Content; import org.apache.nutch.scoring.ScoringFilterException; import org.apache.nutch.scoring.ScoringFilters; -import org.apache.nutch.util.*; +import org.apache.hadoop.conf.Configuration; +import org.apache.hadoop.fs.Path; +import org.apache.hadoop.io.Text; +import org.apache.hadoop.io.Writable; +import org.apache.hadoop.io.WritableComparable; +import java.io.File; +import java.io.IOException; import java.lang.invoke.MethodHandles; import java.text.SimpleDateFormat; -import java.util.*; +import java.util.ArrayList; +import java.util.Arrays; import java.util.HashMap; import java.util.Iterator; import java.util.Map; +import java.util.Map.Entry; /* Parse content in a segment. */ @@ -76,7 +86,6 @@ public ParseSegment(Configuration conf) { @Override public void setup(Mapper<WritableComparable<?>, Content, Text, ParseImpl>.Context context) { Configuration conf = context.getConfiguration(); - //setConf(conf); scfilters = new ScoringFilters(conf); skipTruncated = conf.getBoolean(SKIP_TRUNCATED, true); } @@ -213,8 +222,8 @@ public static boolean isTruncated(Content content) { public void reduce(Text key, Iterable<Writable> values, Context context) throws IOException, InterruptedException { - Iterator<Writable> values_iter = values.iterator(); - context.write(key, values_iter.next()); // collect first value + Iterator<Writable> valuesIter = values.iterator(); + context.write(key, valuesIter.next()); // collect first value } } @@ -229,8 +238,8 @@ public void parse(Path segment) throws IOException, SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss"); long start = System.currentTimeMillis(); if (LOG.isInfoEnabled()) { - LOG.info("ParseSegment: starting at " + sdf.format(start)); - LOG.info("ParseSegment: segment: " + segment); + LOG.info("ParseSegment: starting at {}", sdf.format(start)); + LOG.info("ParseSegment: segment: {}", segment); } Job job = NutchJob.getInstance(getConf()); diff --git a/src/java/org/apache/nutch/parse/ParseText.java b/src/java/org/apache/nutch/parse/ParseText.java index cc6ceee62..024911cc7 100644 --- a/src/java/org/apache/nutch/parse/ParseText.java +++ b/src/java/org/apache/nutch/parse/ParseText.java @@ -17,11 +17,20 @@ package org.apache.nutch.parse; -import java.io.*; -import org.apache.hadoop.io.*; -import org.apache.hadoop.conf.*; -import org.apache.hadoop.util.*; -import org.apache.hadoop.fs.*; +import org.apache.hadoop.conf.Configuration; +import org.apache.hadoop.fs.FileSystem; +import org.apache.hadoop.fs.Path; +import org.apache.hadoop.io.ArrayFile; +import org.apache.hadoop.io.Text; +import org.apache.hadoop.io.VersionMismatchException; +import org.apache.hadoop.io.Writable; +import org.apache.hadoop.io.WritableUtils; +import org.apache.hadoop.util.GenericOptionsParser; + +import java.io.DataInput; +import java.io.DataOutput; +import java.io.IOException; +import org.apache.commons.cli.Options; import org.apache.nutch.util.NutchConfiguration; @@ -31,9 +40,10 @@ public final class ParseText implements Writable

{ public static final String DIR_NAME = "parse_text"; - private final static byte VERSION = 2; + private static final byte VERSION = 2; public ParseText() { + //default constructor } private String text; @@ -61,7 +71,7 @@ public final void write(DataOutput out) throws IOException { Text.writeString(out, text); } - public final static ParseText read(DataInput in) throws IOException { + public static final ParseText read(DataInput in) throws IOException { ParseText parseText = new ParseText(); parseText.readFields(in); return parseText; @@ -74,6 +84,7 @@ public String getText() { return text; } + @Override public boolean equals(Object o) { if (!(o instanceof ParseText)) return false; @@ -81,6 +92,7 @@ public boolean equals(Object o) { return this.text.equals(other.text); } + @Override public String toString() { return text; } diff --git a/src/java/org/apache/nutch/parse/ParseUtil.java b/src/java/org/apache/nutch/parse/ParseUtil.java index bd7380dcd..fd933b468 100644 --- a/src/java/org/apache/nutch/parse/ParseUtil.java +++ b/src/java/org/apache/nutch/parse/ParseUtil.java @@ -16,8 +16,6 @@ */ package org.apache.nutch.parse; -// Commons Logging imports - import java.lang.invoke.MethodHandles; import java.util.concurrent.ExecutorService; import java.util.concurrent.Executors; diff --git a/src/java/org/apache/nutch/parse/Parser.java b/src/java/org/apache/nutch/parse/Parser.java index d101453da..c86a958e9 100644 --- a/src/java/org/apache/nutch/parse/Parser.java +++ b/src/java/org/apache/nutch/parse/Parser.java @@ -17,10 +17,8 @@ package org.apache.nutch.parse; -// Hadoop imports import org.apache.hadoop.conf.Configurable; -// Nutch imports import org.apache.nutch.plugin.Pluggable; import org.apache.nutch.protocol.Content; diff --git a/src/java/org/apache/nutch/parse/ParserFactory.java b/src/java/org/apache/nutch/parse/ParserFactory.java index 7e0d960a7..c68c26ff7 100644 --- a/src/java/org/apache/nutch/parse/ParserFactory.java +++ b/src/java/org/apache/nutch/parse/ParserFactory.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.parse; -// JDK imports import java.lang.invoke.MethodHandles; import java.util.ArrayList; import java.util.Collections; @@ -24,14 +23,11 @@ import java.util.List; import java.util.Vector; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Hadoop imports import org.apache.hadoop.conf.Configuration; -// Nutch imports import org.apache.nutch.plugin.Extension; import org.apache.nutch.plugin.ExtensionPoint; import org.apache.nutch.plugin.PluginRuntimeException; diff --git a/src/java/org/apache/nutch/protocol/Content.java b/src/java/org/apache/nutch/protocol/Content.java index 159abe5e8..7de491617 100755 --- a/src/java/org/apache/nutch/protocol/Content.java +++ b/src/java/org/apache/nutch/protocol/Content.java @@ -17,7 +17,6 @@ package org.apache.nutch.protocol; -//JDK imports import java.io.ByteArrayInputStream; import java.io.DataInput; import java.io.DataInputStream; @@ -26,7 +25,6 @@ import java.util.Arrays; import java.util.zip.InflaterInputStream; -//Hadoop imports import org.apache.commons.cli.Options; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.fs.FileSystem; @@ -37,7 +35,6 @@ import org.apache.hadoop.io.Writable; import org.apache.hadoop.util.GenericOptionsParser; -//Nutch imports import org.apache.nutch.metadata.Metadata; import org.apache.nutch.util.MimeUtil; import org.apache.nutch.util.NutchConfiguration; diff --git a/src/java/org/apache/nutch/protocol/Protocol.java b/src/java/org/apache/nutch/protocol/Protocol.java index ddebffba5..9835744b1 100644 --- a/src/java/org/apache/nutch/protocol/Protocol.java +++ b/src/java/org/apache/nutch/protocol/Protocol.java @@ -19,11 +19,9 @@ import java.util.List; -// Hadoop imports import org.apache.hadoop.conf.Configurable; import org.apache.hadoop.io.Text; -// Nutch imports import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.plugin.Pluggable; diff --git a/src/java/org/apache/nutch/protocol/ProtocolFactory.java b/src/java/org/apache/nutch/protocol/ProtocolFactory.java index 8da0158d4..6a4205931 100644 --- a/src/java/org/apache/nutch/protocol/ProtocolFactory.java +++ b/src/java/org/apache/nutch/protocol/ProtocolFactory.java @@ -23,8 +23,10 @@ import org.slf4j.Logger; import org.slf4j.LoggerFactory; - -import org.apache.nutch.plugin.*; +import org.apache.nutch.plugin.Extension; +import org.apache.nutch.plugin.ExtensionPoint; +import org.apache.nutch.plugin.PluginRepository; +import org.apache.nutch.plugin.PluginRuntimeException; import org.apache.nutch.util.ObjectCache; import org.apache.hadoop.conf.Configuration; diff --git a/src/java/org/apache/nutch/protocol/RobotRulesParser.java b/src/java/org/apache/nutch/protocol/RobotRulesParser.java index 2597147d3..a5a2d975e 100644 --- a/src/java/org/apache/nutch/protocol/RobotRulesParser.java +++ b/src/java/org/apache/nutch/protocol/RobotRulesParser.java @@ -17,7 +17,6 @@ package org.apache.nutch.protocol; -// JDK imports import java.io.File; import java.io.FileReader; import java.io.IOException; @@ -33,11 +32,9 @@ import java.util.Set; import java.util.StringTokenizer; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Nutch imports import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.io.Text; import org.apache.hadoop.util.StringUtils; diff --git a/src/java/org/apache/nutch/scoring/webgraph/LinkDumper.java b/src/java/org/apache/nutch/scoring/webgraph/LinkDumper.java index 663ac52c4..e134441dd 100644 --- a/src/java/org/apache/nutch/scoring/webgraph/LinkDumper.java +++ b/src/java/org/apache/nutch/scoring/webgraph/LinkDumper.java @@ -22,11 +22,8 @@ import java.lang.invoke.MethodHandles; import java.text.SimpleDateFormat; import java.util.ArrayList; -import java.util.Iterator; import java.util.List; import java.util.Random; -import java.util.Set; - import org.apache.commons.cli.CommandLine; import org.apache.commons.cli.CommandLineParser; import org.apache.commons.cli.GnuParser; @@ -48,7 +45,6 @@ import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.Mapper.Context; -import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import

org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; diff --git
a/src/java/org/apache/nutch/scoring/webgraph/LinkRank.java
b/src/java/org/apache/nutch/scoring/webgraph/LinkRank.java index 67643d458..6b964eb0e 100644 ---
a/src/java/org/apache/nutch/scoring/webgraph/LinkRank.java +++
b/src/java/org/apache/nutch/scoring/webgraph/LinkRank.java @@ -23,7 +23,6 @@ import
java.text.SimpleDateFormat; import java.util.ArrayList; import java.util.HashSet; -import java.util.Iterator; import
java.util.List; import java.util.Random; import java.util.Set; @@ -52,7 +51,6 @@ import
org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; import
org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import
org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; diff --git
a/src/java/org/apache/nutch/scoring/webgraph/NodeDumper.java
b/src/java/org/apache/nutch/scoring/webgraph/NodeDumper.java index 459b30682..4e98fcb13 100644 ---
a/src/java/org/apache/nutch/scoring/webgraph/NodeDumper.java +++
b/src/java/org/apache/nutch/scoring/webgraph/NodeDumper.java @@ -19,7 +19,6 @@ import java.io.IOException;
import java.lang.invoke.MethodHandles; import java.text.SimpleDateFormat; -import java.util.Iterator; import
org.apache.commons.cli.CommandLine; import org.apache.commons.cli.CommandLineParser; @@ -40,7 +39,6
@@ import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import
org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; diff --git
a/src/java/org/apache/nutch/scoring/webgraph/NodeReader.java
b/src/java/org/apache/nutch/scoring/webgraph/NodeReader.java index 81ac9df1a..b90cfe581 100644 ---
a/src/java/org/apache/nutch/scoring/webgraph/NodeReader.java +++
b/src/java/org/apache/nutch/scoring/webgraph/NodeReader.java @@ -27,7 +27,6 @@ import
org.apache.commons.cli.Options; import org.apache.hadoop.conf.Configuration; import
org.apache.hadoop.conf.Configured; -import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.MapFile; import org.apache.hadoop.io.Text; diff --git
a/src/java/org/apache/nutch/scoring/webgraph/ScoreUpdater.java
b/src/java/org/apache/nutch/scoring/webgraph/ScoreUpdater.java index a991119a8..a5bb3f348 100644 ---
a/src/java/org/apache/nutch/scoring/webgraph/ScoreUpdater.java +++
b/src/java/org/apache/nutch/scoring/webgraph/ScoreUpdater.java @@ -19,7 +19,6 @@ import java.io.IOException;
import java.lang.invoke.MethodHandles; import java.text.SimpleDateFormat; -import java.util.Iterator; import
java.util.Random; import org.apache.commons.cli.CommandLine; @@ -43,7 +42,6 @@ import
org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; import
org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import
org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat;
import org.apache.hadoop.util.StringUtils; diff --git a/src/java/org/apache/nutch/scoring/webgraph/WebGraph.java
b/src/java/org/apache/nutch/scoring/webgraph/WebGraph.java index 39621ca98..cfef442c8 100644 ---
a/src/java/org/apache/nutch/scoring/webgraph/WebGraph.java +++
b/src/java/org/apache/nutch/scoring/webgraph/WebGraph.java @@ -21,7 +21,6 @@ import
java.text.SimpleDateFormat; import java.util.ArrayList; import java.util.HashSet; -import java.util.Iterator; import
java.util.LinkedHashMap; import java.util.List; import java.util.Map; @@ -51,7 +50,6 @@ import
org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.lib.output.MapFileOutputFormat; import
org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import
org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat;
import org.apache.hadoop.util.StringUtils; diff --git a/src/java/org/apache/nutch/segment/SegmentReader.java
b/src/java/org/apache/nutch/segment/SegmentReader.java index 1421543e2..28b88cd6e 100644 ---
a/src/java/org/apache/nutch/segment/SegmentReader.java +++
b/src/java/org/apache/nutch/segment/SegmentReader.java @@ -57,8 +57,6 @@ import
org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.TaskAttemptContext; import
org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; -import
org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; -import org.apache.hadoop.util.Progressable;
import org.apache.hadoop.util.Tool; import org.apache.hadoop.util.ToolRunner; import
org.apache.nutch.crawl.CrawlDatum; @@ -77,7 +75,12 @@ private static final Logger LOG = LoggerFactory
.getLogger(MethodHandles.lookup().lookupClass()); - private boolean co, fe, ge, pa, pd, pt; + private boolean co; +
private boolean fe; + private boolean ge; + private boolean pa; + private boolean pd; + private boolean pt; public
static class InputCompatMapper extends Mapper<WritableComparable<?>, Writable, Text, NutchWritable> { diff --
git a/src/java/org/apache/nutch/service/NutchReader.java b/src/java/org/apache/nutch/service/NutchReader.java
index ecc870554..d988b697f 100644 --- a/src/java/org/apache/nutch/service/NutchReader.java +++
b/src/java/org/apache/nutch/service/NutchReader.java @@ -21,7 +21,6 @@ import java.util.List; import
org.apache.hadoop.conf.Configuration; -import org.apache.nutch.service.impl.SequenceReader; import
org.apache.nutch.util.NutchConfiguration; import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git
a/src/java/org/apache/nutch/service/impl/NodeReader.java

b/src/java/org/apache/nutch/service/impl/NodeReader.java index 3e5bfba3b..28d6600d1 100644 --- a/src/java/org/apache/nutch/service/impl/NodeReader.java +++ b/src/java/org/apache/nutch/service/impl/NodeReader.java @@ -37,7 +37,6 @@ @Override public List read(String path) throws FileNotFoundException { - // TODO Auto-generated method stub List<HashMap> rows= new ArrayList<>(); Path file = new Path(path); SequenceFile.Reader reader; diff --git a/src/java/org/apache/nutch/service/impl/NutchServerPoolExecutor.java b/src/java/org/apache/nutch/service/impl/NutchServerPoolExecutor.java index 3fc5ba310..147b61aed 100644 --- a/src/java/org/apache/nutch/service/impl/NutchServerPoolExecutor.java +++ b/src/java/org/apache/nutch/service/impl/NutchServerPoolExecutor.java @@ -30,8 +30,6 @@ import com.google.common.collect.Lists; import com.google.common.collect.Queues; - - public class NutchServerPoolExecutor extends ThreadPoolExecutor{ private Queue<JobWorker> workersHistory; diff --git a/src/java/org/apache/nutch/service/model/request/JobConfig.java b/src/java/org/apache/nutch/service/model/request/JobConfig.java index 9cb862c5e..1088ab70c 100644 --- a/src/java/org/apache/nutch/service/model/request/JobConfig.java +++ b/src/java/org/apache/nutch/service/model/request/JobConfig.java @@ -21,7 +21,6 @@ import org.apache.nutch.service.JobManager.JobType; - public class JobConfig { private String crawlId; private JobType type; diff --git a/src/java/org/apache/nutch/service/resources/ConfigResource.java b/src/java/org/apache/nutch/service/resources/ConfigResource.java index 6afd62167..e625c2018 100644 --- a/src/java/org/apache/nutch/service/resources/ConfigResource.java +++ b/src/java/org/apache/nutch/service/resources/ConfigResource.java @@ -29,11 +29,8 @@ import javax.ws.rs.Path; import javax.ws.rs.PathParam; import javax.ws.rs.Produces; -import javax.ws.rs.WebApplicationException; import javax.ws.rs.core.MediaType; import javax.ws.rs.core.Response; -import javax.ws.rs.core.Response.Status; - import org.apache.nutch.service.model.request.NutchConfig; import com.fasterxml.jackson.jaxrs.annotation.JacksonFeatures; import com.fasterxml.jackson.databind.SerializationFeature; diff --git a/src/java/org/apache/nutch/service/resources/SeedResource.java b/src/java/org/apache/nutch/service/resources/SeedResource.java index e8a5be31d..8489e3eb8 100644 --- a/src/java/org/apache/nutch/service/resources/SeedResource.java +++ b/src/java/org/apache/nutch/service/resources/SeedResource.java @@ -39,7 +39,6 @@ import org.slf4j.Logger; import org.slf4j.LoggerFactory; - @Path("/seed") public class SeedResource extends AbstractResource { private static final Logger LOG = LoggerFactory diff --git a/src/java/org/apache/nutch/tools/AbstractCommonCrawlFormat.java b/src/java/org/apache/nutch/tools/AbstractCommonCrawlFormat.java index 21fa23ad5..f693a9780 100644 --- a/src/java/org/apache/nutch/tools/AbstractCommonCrawlFormat.java +++ b/src/java/org/apache/nutch/tools/AbstractCommonCrawlFormat.java @@ -18,10 +18,8 @@ package org.apache.nutch.tools; import java.io.IOException; -import java.io.UnsupportedEncodingException; import java.lang.invoke.MethodHandles; import java.net.InetAddress; -import java.net.URLEncoder; import java.net.UnknownHostException; import java.text.ParseException; import java.util.List; diff --git a/src/java/org/apache/nutch/tools/CommonCrawlDataDumper.java b/src/java/org/apache/nutch/tools/CommonCrawlDataDumper.java index 107ec1c52..3fbe2a7a6 100644 --- a/src/java/org/apache/nutch/tools/CommonCrawlDataDumper.java +++ b/src/java/org/apache/nutch/tools/CommonCrawlDataDumper.java @@ -17,8 +17,6 @@ package org.apache.nutch.tools; -//JDK imports - import java.io.BufferedOutputStream; import java.io.ByteArrayInputStream; import java.io.ByteArrayOutputStream; @@ -51,11 +49,9 @@ import org.apache.commons.compress.archivers.tar.TarArchiveEntry; import org.apache.commons.compress.archivers.tar.TarArchiveOutputStream; import org.apache.commons.compress.compressors.gzip.GzipCompressorOutputStream; -//Commons imports import org.apache.commons.io.IOUtils; import org.apache.commons.io.FilenameUtils; -//Hadoop import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.conf.Configured; import org.apache.hadoop.fs.FileSystem; @@ -75,7 +71,6 @@ import org.apache.nutch.util.DumpFileUtil; import org.apache.nutch.util.NutchConfiguration; -//Tika imports import org.apache.tika.Tika; import com.fasterxml.jackson.dataformat.cbor.CBORFactory; diff --git a/src/java/org/apache/nutch/tools/CommonCrawlFormatWARC.java b/src/java/org/apache/nutch/tools/CommonCrawlFormatWARC.java index 191e42e0d..b79336150 100644 --- a/src/java/org/apache/nutch/tools/CommonCrawlFormatWARC.java +++ b/src/java/org/apache/nutch/tools/CommonCrawlFormatWARC.java @@ -8,11 +8,9 @@ import java.text.ParseException; import java.util.Arrays; import java.util.Collections; -import java.util.Date; import java.util.List; import java.util.concurrent.atomic.AtomicInteger; -import com.ibm.icu.text.SimpleDateFormat; import org.apache.commons.lang.NotImplementedException; import org.apache.commons.lang.StringUtils; import org.apache.hadoop.conf.Configuration; @@ -36,7 +34,7 @@ public static final String TEMPLATE = "${prefix}-${timestamp17}-${serialno}"; private static final AtomicInteger SERIALNO = new AtomicInteger(); - private final static UUIDGenerator GENERATOR = new UUIDGenerator(); + private final static UUIDGenerator GENERATOR = new UUIDGenerator(); private String outputDir = null; private ByteArrayOutputStream out; diff --git a/src/java/org/apache/nutch/tools/DmozParser.java b/src/java/org/apache/nutch/tools/DmozParser.java index

ca9580957..217a15e57 100644 --- a/src/java/org/apache/nutch/tools/DmozParser.java +++ b/src/java/org/apache/nutch/tools/DmozParser.java @@ -17,26 +17,43 @@ package org.apache.nutch.tools; -import java.io.*; +import java.io.BufferedInputStream; +import java.io.BufferedReader; +import java.io.File; +import java.io.FileInputStream; +import java.io.FilterReader; +import java.io.IOException; +import java.io.InputStreamReader; +import java.io.Reader; import java.lang.invoke.MethodHandles; -import java.util.*; -import java.util.regex.*; +import java.util.Random; +import java.util.Vector; +import java.util.regex.Pattern; + +import javax.xml.parsers.ParserConfigurationException; +import javax.xml.parsers.SAXParser; +import javax.xml.parsers.SAXParserFactory; -import javax.xml.parsers.*; -import org.xml.sax.*; -import org.xml.sax.helpers.*; import org.apache.xerces.util.XMLChar; -// Slf4j Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; - -import org.apache.hadoop.io.*; -import org.apache.hadoop.fs.*; +import org.xml.sax.Attributes; +import org.xml.sax.InputSource; +import org.xml.sax.Locator; +import org.xml.sax.SAXException; +import org.xml.sax.SAXParseException; +import org.xml.sax.XMLReader; +import org.xml.sax.helpers.DefaultHandler; import org.apache.hadoop.conf.Configuration; +import org.apache.hadoop.fs.FileSystem; +import org.apache.hadoop.io.MD5Hash; import org.apache.nutch.util.NutchConfiguration; -/** Utility that converts DMOZ RDF into a flat file of URLs to be injected. */ +/** + * Utility that converts DMOZ + * RDF into a flat file of URLs to be injected. + */ public class DmozParser { private static final Logger LOG = LoggerFactory .getLogger(MethodHandles.lookup().lookupClass()); @@ -267,9 +284,8 @@ public void warning(SAXParseException spe) { * the web db. */ public void parseDmozFile(File dmozFile, int subsetDenom, - boolean includeAdult, int skew, Pattern topicPattern) - throws IOException, SAXException, ParserConfigurationException { + boolean includeAdult, int skew, Pattern topicPattern) + throws IOException, SAXException, ParserConfigurationException { SAXParserFactory parserFactory = SAXParserFactory.newInstance(); SAXParser parser = parserFactory.newSAXParser(); @@ -306,7 +322,7 @@ private static void addTopicsFromFile(String topicFile, Vector<String> topics) topicFile), "UTF-8"))) { String line = null; while ((line = in.readLine()) != null) { - topics.addElement(new String(line)); + topics.addElement(line); } } catch (Exception e) { if (LOG.isErrorEnabled()) { @@ -321,7 +337,7 @@ private static void addTopicsFromFile(String topicFile, Vector<String> topics) * structured DMOZ file. By default, we ignore Adult material (as categorized * by DMOZ). */ - public static void main(String argv[]) throws Exception { + public static void main(String[] argv) throws Exception { if (argv.length < 1) { System.err .println("Usage: DmozParser <dmoz_file> [-subset <subsetDenominator>] [-includeAdultMaterial] [-skew skew] [-topicFile <topic list file>] [-topic <topic> [-topic <topic> [...]]]"); @@ -362,7 +378,7 @@ public static void main(String argv[]) throws Exception { DmozParser parser = new DmozParser(); if (!topics.isEmpty()) { - String regExp = new String("^("); + String regExp = "^("; int j = 0; for (; j < topics.size() - 1; ++j) { regExp = regExp.concat(topics.get(j)); diff --git a/src/java/org/apache/nutch/tools/FileDumper.java b/src/java/org/apache/nutch/tools/FileDumper.java index 51cc12455..fcf2f199b 100644 --- a/src/java/org/apache/nutch/tools/FileDumper.java +++ b/src/java/org/apache/nutch/tools/FileDumper.java @@ -17,10 +17,8 @@ package org.apache.nutch.tools; -//JDK imports import java.io.DataOutputStream; import java.io.File; -import java.io.FileFilter; import java.io.FileOutputStream; import java.io.ByteArrayInputStream; import java.lang.invoke.MethodHandles; @@ -35,12 +33,10 @@ import org.apache.commons.cli.Option; import org.apache.commons.cli.OptionBuilder; import org.apache.commons.cli.Options; -//Commons imports import org.apache.commons.io.IOUtils; import org.apache.commons.io.FilenameUtils; import org.apache.commons.codec.digest.DigestUtils; -//Hadoop import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.fs.Path; @@ -52,7 +48,6 @@ import org.apache.nutch.util.NutchConfiguration; import org.apache.nutch.util.TableUtil; -//Tika imports import org.apache.tika.Tika; import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/java/org/apache/nutch/tools/FreeGenerator.java b/src/java/org/apache/nutch/tools/FreeGenerator.java index fef1a0bae..3b01bb431 100644 --- a/src/java/org/apache/nutch/tools/FreeGenerator.java +++ b/src/java/org/apache/nutch/tools/FreeGenerator.java @@ -21,7 +21,6 @@ import java.lang.invoke.MethodHandles; import java.text.SimpleDateFormat; import java.util.HashMap; -import java.util.Iterator; import java.util.Map.Entry; import org.slf4j.Logger; @@ -35,7 +34,6 @@ import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat; import org.apache.hadoop.mapreduce.lib.input.TextInputFormat; diff --git a/src/java/org/apache/nutch/tools/WARCUtils.java b/src/java/org/apache/nutch/tools/WARCUtils.java index d8ae0b381..a705ae7c1 100644 --- a/src/java/org/apache/nutch/tools/WARCUtils.java +++ b/src/java/org/apache/nutch/tools/WARCUtils.java @@ -20,135 +20,135 @@ import org.archive.util.anvl.ANVLRecord; public class WARCUtils { - public final static String SOFTWARE = "software"; - public final static String HTTP_HEADER_FROM = "http-header-from"; - public final static String HTTP_HEADER_USER_AGENT = "http-header-user-agent"; - public final static String HOSTNAME = "hostname"; - public final static String ROBOTS = "robots"; - public final static String OPERATOR = "operator"; - public final static String FORMAT = "format"; - public final static String CONFORMS_TO = "conformsTo"; - public final static String IP = "ip"; - public final static UUIDGenerator generator = new UUIDGenerator(); - - public final static ANVLRecord getWARCInfoContent(Configuration conf) { - ANVLRecord record = new

ANVLRecord(); - - // informative headers - record.addLabelValue(FORMAT, "WARC File Format 1.0"); - record.addLabelValue(CONFORMS_TO, "http://bibnum.bnf.fr/WARC/WARC_ISO_28500_version1_latestdraft.pdf"); - - record.addLabelValue(SOFTWARE, conf.get("http.agent.name", "")); - record.addLabelValue(HTTP_HEADER_USER_AGENT, - getAgentString(conf.get("http.agent.name", ""), - conf.get("http.agent.version", ""), - conf.get("http.agent.description", ""), - conf.get("http.agent.url", ""), - conf.get("http.agent.email", ""))); - record.addLabelValue(HTTP_HEADER_FROM, - conf.get("http.agent.email", "")); - - try { - record.addLabelValue(HOSTNAME, getHostname(conf)); - record.addLabelValue(IP, getIPAddress(conf)); - } catch (UnknownHostException ignored) { - // do nothing as this fields are optional - } - - record.addLabelValue(ROBOTS, "classic"); // TODO Make configurable? - record.addLabelValue(OPERATOR, conf.get("http.agent.email", "")); - - return record; + public final static String SOFTWARE = "software"; + public final static String HTTP_HEADER_FROM = "http-header-from"; + public final static String HTTP_HEADER_USER_AGENT = "http-header-user-agent"; + public final static String HOSTNAME = "hostname"; + public final static String ROBOTS = "robots"; + public final static String OPERATOR = "operator"; + public final static String FORMAT = "format"; + public final static String CONFORMS_TO = "conformsTo"; + public final static String IP = "ip"; + public final static UUIDGenerator generator = new UUIDGenerator(); + + public static final ANVLRecord getWARCInfoContent(Configuration conf) { + ANVLRecord record = new ANVLRecord(); + + // informative headers + record.addLabelValue(FORMAT, "WARC File Format 1.0"); + record.addLabelValue(CONFORMS_TO, "http://bibnum.bnf.fr/WARC/WARC_ISO_28500_version1_latestdraft.pdf"); + + record.addLabelValue(SOFTWARE, conf.get("http.agent.name", "")); + record.addLabelValue(HTTP_HEADER_USER_AGENT, + getAgentString(conf.get("http.agent.name", ""), + conf.get("http.agent.version", ""), + conf.get("http.agent.description", ""), + conf.get("http.agent.url", ""), + conf.get("http.agent.email", ""))); + record.addLabelValue(HTTP_HEADER_FROM, + conf.get("http.agent.email", "")); + + try { + record.addLabelValue(HOSTNAME, getHostname(conf)); + record.addLabelValue(IP, getIPAddress(conf)); + } catch (UnknownHostException ignored) { + // do nothing as this fields are optional + } - public static final String getHostname(Configuration conf) - throws UnknownHostException { + record.addLabelValue(ROBOTS, "classic"); // TODO Make configurable? + record.addLabelValue(OPERATOR, conf.get("http.agent.email", "")); - return StringUtil.isEmpty(conf.get("http.agent.host", "")) ? - InetAddress.getLocalHost().getHostName() : - conf.get("http.agent.host"); - } + return record; + } - public static final String getIPAddress(Configuration conf) - throws UnknownHostException { + public static final String getHostname(Configuration conf) + throws UnknownHostException { - return InetAddress.getLocalHost().getHostAddress(); - } + return StringUtil.isEmpty(conf.get("http.agent.host", "")) ? + InetAddress.getLocalHost().getHostName() : + conf.get("http.agent.host"); + } - public static final byte[] toByteArray(HttpHeaders headers) - throws IOException { - ByteArrayOutputStream out = new ByteArrayOutputStream(); - headers.write(out); + public static final String getIPAddress(Configuration conf) + throws UnknownHostException { - return out.toByteArray(); - } + return InetAddress.getLocalHost().getHostAddress(); + } - public static final String getAgentString(String name, String version, - String description, String URL, String email) { + public static final byte[] toByteArray(HttpHeaders headers) + throws IOException { + ByteArrayOutputStream out = new ByteArrayOutputStream(); + headers.write(out); - StringBuffer buf = new StringBuffer(); + return out.toByteArray(); + } - buf.append(name); + public static final String getAgentString(String name, String version, + String description, String URL, String email) { - if (version != null) { - buf.append("/").append(version); - } + StringBuffer buf = new StringBuffer(); - if (((description != null) && (description.length() != 0)) || (- (email != null) && (email.length() != 0)) || ((URL != null) && (- URL.length() != 0))) { - buf.append(" ("); + buf.append(name); - if ((description != null) && (description.length() != 0)) { - buf.append(description); - if ((URL != null) || (email != null)) - buf.append("; "); - } + if (version != null) { + buf.append("/").append(version); + } - if ((URL != null) && (URL.length() != 0)) { - buf.append(URL); - if (email != null) - buf.append("; "); - } + if (((description != null) && (description.length() != 0)) || (+ (email != null) && (email.length() != 0)) || ((URL != null) && (+ URL.length() != 0))) { + buf.append(" ("); - if ((email != null) && (email.length() != 0)) - buf.append(email); + if ((description != null) && (description.length() != 0)) { + buf.append(description); + if ((URL != null) || (email != null)) + buf.append("; "); + } - buf.append(")"); - } + if ((URL != null) && (URL.length() != 0)) { + buf.append(URL); + if (email != null) + buf.append("; "); + } - return buf.toString(); + if ((email != null) && (email.length() != 0)) + buf.append(email); + + buf.append(")"); } - public static final WARCRecordInfo docToMetadata(NutchDocument doc) - throws UnsupportedEncodingException { - WARCRecordInfo record = new WARCRecordInfo(); - - record.setType(WARCConstants.WARCRecordType.metadata); - record.setUrl((String) doc.getFieldValue("id")); - record.setCreate14DigitDate(- DateUtils.get14DigitDate((Date) doc.getFieldValue("tstamp"))); - record.setMimetype("application/warc-fields"); - record.setRecordId(generator.getRecordID()); - - // metadata - ANVLRecord metadata = new ANVLRecord(); - - for (String field : doc.getFieldNames()) { - List<Object> values = doc.getField(field).getValues(); - for (Object value : values) { - if (value instanceof Date) { - metadata.addLabelValue(field, DateUtils.get14DigitDate()); - } else { - metadata.addLabelValue(field, (String) value); - } - } + return buf.toString(); + } + + public static final WARCRecordInfo docToMetadata(NutchDocument doc) + throws UnsupportedEncodingException { + WARCRecordInfo record = new WARCRecordInfo(); + + record.setType(WARCConstants.WARCRecordType.metadata); + record.setUrl((String) doc.getFieldValue("id")); +

record.setCreate14DigitDate(+ DateUtils.get14DigitDate((Date) doc.getFieldValue("tstamp"))); + record.setMimetype("application/warc-fields"); + record.setRecordId(generator.getRecordID()); + + // metadata + ANVLRecord metadata = new ANVLRecord(); + + for (String field : doc.getFieldNames()) { + List<Object> values = doc.getField(field).getValues(); + for (Object value : values) { + if (value instanceof Date) { + metadata.addLabelValue(field, DateUtils.get14DigitDate()); + } else { + metadata.addLabelValue(field, (String) value); } + } + } - record.setContentLength(metadata.getLength()); - record.setContentStream(- new ByteArrayInputStream(metadata.getUTF8Bytes())); + record.setContentLength(metadata.getLength()); + record.setContentStream(+ new ByteArrayInputStream(metadata.getUTF8Bytes())); - return record; - } + return record; + } } diff --git a/src/java/org/apache/nutch/tools/arc/ArcSegmentCreator.java b/src/java/org/apache/nutch/tools/arc/ArcSegmentCreator.java index 9599f933f..1f9e660f6 100644 --- a/src/java/org/apache/nutch/tools/arc/ArcSegmentCreator.java +++ b/src/java/org/apache/nutch/tools/arc/ArcSegmentCreator.java @@ -33,7 +33,6 @@ import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.util.StringUtils; import org.apache.hadoop.util.Tool; import org.apache.hadoop.util.ToolRunner; diff --git a/src/java/org/apache/nutch/tools/warc/WARCExporter.java b/src/java/org/apache/nutch/tools/warc/WARCExporter.java index 559f75d3f..aae8064f8 100644 --- a/src/java/org/apache/nutch/tools/warc/WARCExporter.java +++ b/src/java/org/apache/nutch/tools/warc/WARCExporter.java @@ -27,7 +27,6 @@ import java.text.SimpleDateFormat; import java.util.ArrayList; import java.util.Date; -import java.util.Iterator; import java.util.List; import java.util.Locale; import java.util.UUID; @@ -45,7 +44,6 @@ import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; -import org.apache.hadoop.mapreduce.Mapper.Context; import org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat; import org.apache.hadoop.util.Tool; diff --git a/src/java/org/apache/nutch/util/DeflateUtils.java b/src/java/org/apache/nutch/util/DeflateUtils.java index a7e7e3276..558762c2e 100644 --- a/src/java/org/apache/nutch/util/DeflateUtils.java +++ b/src/java/org/apache/nutch/util/DeflateUtils.java @@ -24,7 +24,6 @@ import java.util.zip.InflaterInputStream; import java.util.zip.DeflaterOutputStream; -// Slf4j Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/java/org/apache/nutch/util/DomUtil.java b/src/java/org/apache/nutch/util/DomUtil.java index b4f0eac82..24612865b 100644 --- a/src/java/org/apache/nutch/util/DomUtil.java +++ b/src/java/org/apache/nutch/util/DomUtil.java @@ -37,7 +37,6 @@ import org.xml.sax.InputSource; import org.xml.sax.SAXException; -// Slf4j Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/java/org/apache/nutch/util/GZIPUtils.java b/src/java/org/apache/nutch/util/GZIPUtils.java index 6a6a984f6..392eb1394 100644 --- a/src/java/org/apache/nutch/util/GZIPUtils.java +++ b/src/java/org/apache/nutch/util/GZIPUtils.java @@ -24,7 +24,6 @@ import java.util.zip.GZIPInputStream; import java.util.zip.GZIPOutputStream; -// Slf4j Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/java/org/apache/nutch/util/MimeUtil.java b/src/java/org/apache/nutch/util/MimeUtil.java index 7e8fe7d3f..7f7ec0c1b 100644 --- a/src/java/org/apache/nutch/util/MimeUtil.java +++ b/src/java/org/apache/nutch/util/MimeUtil.java @@ -17,16 +17,13 @@ package org.apache.nutch.util; -// JDK imports import java.io.File; import java.io.IOException; import java.io.InputStream; import java.lang.invoke.MethodHandles; -// Hadoop imports import org.apache.hadoop.conf.Configuration; -// Tika imports import org.apache.tika.Tika; import org.apache.tika.io.TikaInputStream; import org.apache.tika.metadata.Metadata; @@ -35,11 +32,9 @@ import org.apache.tika.mime.MimeTypes; import org.apache.tika.mime.MimeTypesFactory; -// Slf4j logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// imported for Javadoc import org.apache.nutch.protocol.ProtocolOutput; /** diff --git a/src/java/org/apache/nutch/util/ProtocolStatusStatistics.java b/src/java/org/apache/nutch/util/ProtocolStatusStatistics.java index 7e241ffdf..c7a852df3 100644 --- a/src/java/org/apache/nutch/util/ProtocolStatusStatistics.java +++ b/src/java/org/apache/nutch/util/ProtocolStatusStatistics.java @@ -20,7 +20,6 @@ import java.io.IOException; import java.lang.invoke.MethodHandles; -import java.net.URL; import java.text.SimpleDateFormat; import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/java/org/apache/nutch/util/URLUtil.java b/src/java/org/apache/nutch/util/URLUtil.java index 08f5236b2..525f14b0d 100644 --- a/src/java/org/apache/nutch/util/URLUtil.java +++ b/src/java/org/apache/nutch/util/URLUtil.java @@ -17,8 +17,10 @@ package org.apache.nutch.util; +import java.net.IDN; import java.net.MalformedURLException; -import java.net.*; +import java.net.URI; +import java.net.URL; import java.util.regex.Pattern; import org.apache.nutch.util.domain.DomainSuffix; diff --git a/src/java/org/apache/nutch/webui/model/NutchConfig.java b/src/java/org/apache/nutch/webui/model/NutchConfig.java index 2106d2322..7a2111e28 100644 --- a/src/java/org/apache/nutch/webui/model/NutchConfig.java +++ b/src/java/org/apache/nutch/webui/model/NutchConfig.java @@ -1,3 +1,19 @@ +/** + * 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. + */ package org.apache.nutch.webui.model; import java.io.Serializable; diff --git a/src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCIndexingFilter.java b/src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCIndexingFilter.java index fea2eab93..cfaac1f99 100644 --- a/src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCIndexingFilter.java +++ b/src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCIndexingFilter.java @@ -36,8 +36,8 @@ import org.slf4j.LoggerFactory; import java.lang.invoke.MethodHandles; -import java.util.*; import java.net.URL; +import java.util.StringTokenizer; import java.net.MalformedURLException; /** Adds basic searchable fields to a document. */ diff --git a/src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCParseFilter.java b/src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCParseFilter.java index 57de6f647..7b20a2861 100644 --- a/src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCParseFilter.java +++ b/src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCParseFilter.java @@ -18,21 +18,36 @@ package org.creativecommons.nutch; import org.apache.nutch.metadata.CreativeCommons; -import org.apache.nutch.parse.*; import org.apache.nutch.protocol.Content; import org.apache.nutch.metadata.Metadata; +import org.apache.nutch.parse.HTMLMetaTags; +import org.apache.nutch.parse.HtmlParseFilter; +import org.apache.nutch.parse.Parse; +import org.apache.nutch.parse.ParseException; +import org.apache.nutch.parse.ParseResult; +import org.apache.nutch.parse.ParseStatus; +import org.apache.nutch.parse.ParseText; import org.apache.hadoop.conf.Configuration; import org.slf4j.Logger; import org.slf4j.LoggerFactory; - -import java.util.*; +import org.w3c.dom.Comment; +import org.w3c.dom.Document; +import org.w3c.dom.DocumentFragment; +import org.w3c.dom.Element; +import org.w3c.dom.Node; +import org.w3c.dom.NodeList; + +import java.io.StringReader; import java.lang.invoke.MethodHandles; -import java.io.*; -import java.net.*; -import javax.xml.parsers.*; +import java.net.MalformedURLException; +import java.net.URL; +import java.util.HashMap; + +import javax.xml.parsers.DocumentBuilder; +import javax.xml.parsers.DocumentBuilderFactory; import org.xml.sax.InputSource; -import org.w3c.dom.*; /** Adds metadata identifying the Creative Commons license used, if any. */ public class CCParseFilter implements HtmlParseFilter { @@ -184,7 +199,6 @@ private void findRdf(String comment) { if (LOG.isWarnEnabled()) { LOG.warn("CC: Failed to parse RDF in " + base + ": " + e); } - // e.printStackTrace(); return; } @@ -219,12 +233,6 @@ private void findRdf(String comment) { if (!CC_NS.equals(predicateElement.getNamespaceURI())) { continue; } - - // add object and predicate to metadata - // metadata.put(object, predicate); - // if (LOG.isInfoEnabled()) { - // LOG.info("CC: found: "+predicate+"="+object); - // } } } @@ -244,7 +252,7 @@ private void findRdf(String comment) { } } - private static final HashMap<String, String> WORK_TYPE_NAMES = new HashMap<String, String>(); + private static final HashMap<String, String> WORK_TYPE_NAMES = new HashMap<>(); static { WORK_TYPE_NAMES.put("http://purl.org/dc/dcmitype/MovingImage", "video"); WORK_TYPE_NAMES.put("http://purl.org/dc/dcmitype/StillImage", "image"); diff --git a/src/plugin/feed/src/java/org/apache/nutch/indexer/feed/FeedIndexingFilter.java b/src/plugin/feed/src/java/org/apache/nutch/indexer/feed/FeedIndexingFilter.java index 76a171834..2a067d73f 100644 --- a/src/plugin/feed/src/java/org/apache/nutch/indexer/feed/FeedIndexingFilter.java +++ b/src/plugin/feed/src/java/org/apache/nutch/indexer/feed/FeedIndexingFilter.java @@ -17,10 +17,8 @@ package org.apache.nutch.indexer.feed; -//JDK imports import java.util.Date; -//APACHE imports import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.io.Text; import org.apache.nutch.crawl.CrawlDatum; diff --git a/src/plugin/feed/src/java/org/apache/nutch/parse/feed/FeedParser.java b/src/plugin/feed/src/java/org/apache/nutch/parse/feed/FeedParser.java index 5ec1dcf7e..feac07043 100644 --- a/src/plugin/feed/src/java/org/apache/nutch/parse/feed/FeedParser.java +++ b/src/plugin/feed/src/java/org/apache/nutch/parse/feed/FeedParser.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.parse.feed; -// JDK imports import java.lang.invoke.MethodHandles; import java.io.ByteArrayInputStream; import java.io.DataInputStream; @@ -26,13 +25,11 @@ import java.util.List; import java.util.Map.Entry; -// APACHE imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.io.Text; import org.apache.hadoop.util.StringUtils; -// import org.apache.nutch.indexer.anchor.AnchorIndexingFilter; removed as per NUTCH-1078 import org.apache.nutch.metadata.Feed; import org.apache.nutch.metadata.Metadata; import org.apache.nutch.net.URLFilters; @@ -52,7 +49,6 @@ import org.apache.nutch.util.NutchConfiguration; import org.xml.sax.InputSource; -// ROME imports import com.rometools.rome.feed.synd.SyndCategory; import com.rometools.rome.feed.synd.SyndContent; import com.rometools.rome.feed.synd.SyndEntry; diff --git a/src/plugin/headings/src/java/org/apache/nutch/parse/headings/HeadingsParseFilter.java b/src/plugin/headings/src/java/org/apache/nutch/parse/headings/HeadingsParseFilter.java index eaa2a7020..beddfc352 100644 --- a/src/plugin/headings/src/java/org/apache/nutch/parse/headings/HeadingsParseFilter.java +++ b/src/plugin/headings/src/java/org/apache/nutch/parse/headings/HeadingsParseFilter.java @@ -19,7 +19,9 @@ import java.util.ArrayList; import java.util.List; -import java.util.regex.*; +import java.util.regex.Matcher; +import java.util.regex.Pattern; +import java.util.regex.PatternSyntaxException; import org.apache.hadoop.conf.Configuration; import org.apache.nutch.parse.HTMLMetaTags;

java.util.regex.Pattern; + import org.apache.hadoop.conf.Configuration; import org.apache.nutch.parse.HTMLMetaTags; import org.apache.nutch.parse.Parse; @@ -27,7 +29,8 @@ import org.apache.nutch.parse.ParseResult; import org.apache.nutch.protocol.Content; import org.apache.nutch.util.NodeWalker; -import org.w3c.dom.*; +import org.w3c.dom.DocumentFragment; +import org.w3c.dom.Node; /** * HtmlParseFilter to retrieve h1 and h2 values from the DOM. @@ -81,7 +84,7 @@ public Configuration getConf() { * Finds the specified element and returns its value */ protected List<String> getElement(DocumentFragment doc, String element) { - List<String> headings = new ArrayList<String>(); + List<String> headings = new ArrayList<>(); NodeWalker walker = new NodeWalker(doc); while (walker.hasNext()) { diff --git a/src/plugin/index-more/src/java/org/apache/nutch/indexer/more/MoreIndexingFilter.java b/src/plugin/index-more/src/java/org/apache/nutch/indexer/more/MoreIndexingFilter.java index 26a7df2c6..3de951ab7 100644 --- a/src/plugin/index-more/src/java/org/apache/nutch/indexer/more/MoreIndexingFilter.java +++ b/src/plugin/index-more/src/java/org/apache/nutch/indexer/more/MoreIndexingFilter.java @@ -46,8 +46,10 @@ import java.io.BufferedReader; import java.io.IOException; import java.util.Date; -import java.util.regex.*; import java.util.HashMap; +import java.util.regex.Matcher; +import java.util.regex.Pattern; +import java.util.regex.PatternSyntaxException; import org.apache.commons.lang.StringUtils; import org.apache.commons.lang.time.DateUtils; diff --git a/src/plugin/index-replace/src/java/org/apache/nutch/indexer/replace/FieldReplacer.java b/src/plugin/index-replace/src/java/org/apache/nutch/indexer/replace/FieldReplacer.java index ddfe24dcc..038229d95 100644 --- a/src/plugin/index-replace/src/java/org/apache/nutch/indexer/replace/FieldReplacer.java +++ b/src/plugin/index-replace/src/java/org/apache/nutch/indexer/replace/FieldReplacer.java @@ -20,8 +20,8 @@ import java.util.regex.Pattern; import java.util.regex.PatternSyntaxException; -import org.apache.commons.logging.Log; -import org.apache.commons.logging.LogFactory; +import org.slf4j.Logger; +import org.slf4j.LoggerFactory; /** * POJO to store a filename, its match pattern and its replacement string. @@ -33,7 +33,7 @@ */ public class FieldReplacer { - private static final Log LOG = LogFactory.getLog(FieldReplacer.class + private static final Logger LOG = LoggerFactory.getLogger(FieldReplacer.class .getName()); private final String fieldName; diff --git a/src/plugin/indexer-elastic-rest/src/java/org/apache/nutch/indexwriter/elasticrest/ElasticRestIndexWriter.java b/src/plugin/indexer-elastic-rest/src/java/org/apache/nutch/indexwriter/elasticrest/ElasticRestIndexWriter.java index 090bf771c..02600618f 100644 --- a/src/plugin/indexer-elastic-rest/src/java/org/apache/nutch/indexwriter/elasticrest/ElasticRestIndexWriter.java +++ b/src/plugin/indexer-elastic-rest/src/java/org/apache/nutch/indexwriter/elasticrest/ElasticRestIndexWriter.java @@ -31,7 +31,6 @@ import org.apache.commons.lang.StringUtils; import org.apache.commons.lang3.exception.ExceptionUtils; import org.apache.hadoop.conf.Configuration; -import org.apache.http.HttpResponse; import org.apache.http.concurrent.BasicFuture; import org.apache.http.conn.ssl.DefaultHostnameVerifier; import org.apache.http.conn.ssl.NoopHostnameVerifier; diff --git a/src/plugin/indexer-elastic/src/java/org/apache/nutch/indexwriter/elastic/ElasticIndexWriter.java b/src/plugin/indexer-elastic/src/java/org/apache/nutch/indexwriter/elastic/ElasticIndexWriter.java index 1f7ed765f..a328b7fbd 100644 --- a/src/plugin/indexer-elastic/src/java/org/apache/nutch/indexwriter/elastic/ElasticIndexWriter.java +++ b/src/plugin/indexer-elastic/src/java/org/apache/nutch/indexwriter/elastic/ElasticIndexWriter.java @@ -45,8 +45,7 @@ import org.elasticsearch.common.settings.Settings; import org.elasticsearch.common.transport.InetSocketTransportAddress; import org.elasticsearch.node.Node; -import org.elasticsearch.transport.client.*; - +import org.elasticsearch.transport.client.PreBuiltTransportClient; import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitIndexWriter.java b/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitIndexWriter.java index c1e96a354..86547b890 100644 --- a/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitIndexWriter.java +++ b/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitIndexWriter.java @@ -30,164 +30,164 @@ import org.slf4j.Logger; import org.slf4j.LoggerFactory; -import java.util.*; +import java.util.Map; import java.util.concurrent.TimeoutException; public class RabbitIndexWriter implements IndexWriter { - private String serverHost; - private int serverPort; - private String serverVirtualHost; - private String serverUsername; - private String serverPassword; + private String serverHost; + private int serverPort; + private String serverVirtualHost; + private String serverUsername; + private String serverPassword; - private String exchangeServer; - private String exchangeType; + private String exchangeServer; + private String exchangeType; - private String queueName; - private boolean queueDurable; - private String queueRoutingKey; + private String queueName; + private boolean queueDurable; + private String queueRoutingKey; - private int commitSize; + private int commitSize; - public static final Logger LOG = LoggerFactory.getLogger(RabbitIndexWriter.class); + public static final Logger LOG = LoggerFactory.getLogger(RabbitIndexWriter.class); - private Configuration config; + private Configuration config; - private RabbitMessage rabbitMessage = new RabbitMessage(); + private RabbitMessage rabbitMessage = new RabbitMessage(); - private Channel channel; - private Connection connection; + private Channel channel; + private Connection connection; - @Override - public Configuration getConf() { - return config; - } + @Override + public Configuration getConf() { + return config; + } - @Override - public void setConf(Configuration conf) { - config = conf; - serverHost = conf.get(RabbitMQConstants.SERVER_HOST, "localhost"); - serverPort = conf.getInt(RabbitMQConstants.SERVER_PORT, 15672); - serverVirtualHost =

conf.get(RabbitMQConstants.SERVER_VIRTUAL_HOST, null); + serverHost =
conf.get(RabbitMQConstants.SERVER_HOST, "localhost"); + serverPort =
conf.getInt(RabbitMQConstants.SERVER_PORT, 15672); + serverVirtualHost =
conf.get(RabbitMQConstants.SERVER_VIRTUAL_HOST, null); - serverUsername =
conf.get(RabbitMQConstants.SERVER_USERNAME, "admin"); - serverPassword =
conf.get(RabbitMQConstants.SERVER_PASSWORD, "admin"); + serverUsername =
conf.get(RabbitMQConstants.SERVER_USERNAME, "admin"); + serverPassword =
conf.get(RabbitMQConstants.SERVER_PASSWORD, "admin"); - exchangeServer =
conf.get(RabbitMQConstants.EXCHANGE_SERVER, "nutch.exchange"); - exchangeType =
conf.get(RabbitMQConstants.EXCHANGE_TYPE, "direct"); + exchangeServer =
conf.get(RabbitMQConstants.EXCHANGE_SERVER, "nutch.exchange"); + exchangeType =
conf.get(RabbitMQConstants.EXCHANGE_TYPE, "direct"); - queueName =
conf.get(RabbitMQConstants.QUEUE_NAME, "nutch.queue"); - queueDurable =
conf.getBoolean(RabbitMQConstants.QUEUE_DURABLE, true); - queueRoutingKey =
conf.get(RabbitMQConstants.QUEUE_ROUTING_KEY, "nutch.key"); + queueName =
conf.get(RabbitMQConstants.QUEUE_NAME, "nutch.queue"); + queueDurable =
conf.getBoolean(RabbitMQConstants.QUEUE_DURABLE, true); + queueRoutingKey =
conf.get(RabbitMQConstants.QUEUE_ROUTING_KEY, "nutch.key"); - commitSize =
conf.getInt(RabbitMQConstants.COMMIT_SIZE, 250); - } + commitSize =
conf.getInt(RabbitMQConstants.COMMIT_SIZE, 250); + } - @Override - public void open(Configuration conf,
String name) throws IOException { - ConnectionFactory factory = new ConnectionFactory(); -
factory.setHost(serverHost); - factory.setPort(serverPort); + @Override + public void open(Configuration conf,
String name) throws IOException { + ConnectionFactory factory = new ConnectionFactory(); +
factory.setHost(serverHost); + factory.setPort(serverPort); - if(serverVirtualHost != null) { -
factory.setVirtualHost(serverVirtualHost); - } + if(serverVirtualHost != null) { +
factory.setVirtualHost(serverVirtualHost); + } - factory.setUsername(serverUsername); -
factory.setPassword(serverPassword); + factory.setUsername(serverUsername); +
factory.setPassword(serverPassword); - try { - connection = factory.newConnection(); - channel =
connection.createChannel(); + try { + connection = factory.newConnection(); + channel =
connection.createChannel(); - channel.exchangeDeclare(exchangeServer, exchangeType, true); -
channel.queueDeclare(queueName, queueDurable, false, false, null); - channel.queueBind(queueName,
exchangeServer, queueRoutingKey); + channel.exchangeDeclare(exchangeServer, exchangeType, true); +
channel.queueDeclare(queueName, queueDurable, false, false, null); + channel.queueBind(queueName,
exchangeServer, queueRoutingKey); - } catch (TimeoutException | IOException ex) { - throw
makeIOException(ex); - } + } catch (TimeoutException | IOException ex) { + throw makeIOException(ex); } - -
@Override - public void update(NutchDocument doc) throws IOException { - RabbitDocument rabbitDocument =
new RabbitDocument(); - - for (final Map.Entry<String, NutchField> e : doc) { -
RabbitDocument.RabbitDocumentField field = new RabbitDocument.RabbitDocumentField(- e.getKey(), -
e.getValue().getWeight(), - e.getValue().getValues()); - rabbitDocument.addField(field); - } -
rabbitDocument.setDocumentBoost(doc.getWeight()); - - rabbitMessage.addDocToUpdate(rabbitDocument); -
if(rabbitMessage.size() >= commitSize) { - commit(); - } + } + + @Override + public void update(NutchDocument
doc) throws IOException { + RabbitDocument rabbitDocument = new RabbitDocument(); + + for (final
Map.Entry<String, NutchField> e : doc) { + RabbitDocument.RabbitDocumentField field = new
RabbitDocument.RabbitDocumentField(+ e.getKey(), + e.getValue().getWeight(), + e.getValue().getValues()); +
rabbitDocument.addField(field); } + rabbitDocument.setDocumentBoost(doc.getWeight()); - @Override - public
void commit() throws IOException { - if (!rabbitMessage.isEmpty()) { - channel.basicPublish(exchangeServer,
queueRoutingKey, null, rabbitMessage.getBytes()); - } - rabbitMessage.clear(); +
rabbitMessage.addDocToUpdate(rabbitDocument); + if(rabbitMessage.size() >= commitSize) { + commit(); } + } -
@Override - public void write(NutchDocument doc) throws IOException { - RabbitDocument rabbitDocument =
new RabbitDocument(); - - for (final Map.Entry<String, NutchField> e : doc) { -
RabbitDocument.RabbitDocumentField field = new RabbitDocument.RabbitDocumentField(- e.getKey(), -
e.getValue().getWeight(), - e.getValue().getValues()); - rabbitDocument.addField(field); - } -
rabbitDocument.setDocumentBoost(doc.getWeight()); - - rabbitMessage.addDocToWrite(rabbitDocument); - -
if(rabbitMessage.size() >= commitSize) { - commit(); - } + @Override + public void commit() throws IOException
{ + if (!rabbitMessage.isEmpty()) { + channel.basicPublish(exchangeServer, queueRoutingKey, null,
rabbitMessage.getBytes()); } - - @Override - public void close() throws IOException { - commit();//TODO: This is
because indexing job never call commit method. It should be fixed. - try { - channel.close(); - connection.close(); - }
catch (IOException | TimeoutException e) { - throw makeIOException(e); - } + rabbitMessage.clear(); + } + +
@Override + public void write(NutchDocument doc) throws IOException { + RabbitDocument rabbitDocument =
new RabbitDocument(); + + for (final Map.Entry<String, NutchField> e : doc) { +
RabbitDocument.RabbitDocumentField field = new RabbitDocument.RabbitDocumentField(+ e.getKey(), +
e.getValue().getWeight(), + e.getValue().getValues()); + rabbitDocument.addField(field); } +
rabbitDocument.setDocumentBoost(doc.getWeight()); - @Override - public void delete(String url) throws
IOException { - rabbitMessage.addDocToDelete(url); + rabbitMessage.addDocToWrite(rabbitDocument); -

if(rabbitMessage.size() >= commitSize) { - commit(); - } + if(rabbitMessage.size() >= commitSize) { + commit(); } - - private static IOException makeIOException(Exception e) { - return new IOException(e); + } + + @Override + public void close() throws IOException { + commit();//TODO: This is because indexing job never call commit method. It should be fixed. + try { + channel.close(); + connection.close(); + } catch (IOException | TimeoutException e) { + throw makeIOException(e); } + } + + @Override + public void delete(String url) throws IOException { + rabbitMessage.addDocToDelete(url); - public String describe() { - return "RabbitIndexWriter\n" + - "\t" + serverHost + ":" + serverPort + " : URL of RabbitMQ server\n" + - "\t" + RabbitMQConstants.SERVER_VIRTUAL_HOST + " : Virtualhost name\n" + - "\t" + RabbitMQConstants.SERVER_USERNAME + " : Username for authentication\n" + - "\t" + RabbitMQConstants.SERVER_PASSWORD + " : Password for authentication\n" + - "\t" + RabbitMQConstants.COMMIT_SIZE + " : Buffer size when sending to RabbitMQ (default 250)\n"; + if(rabbitMessage.size() >= commitSize) { + commit(); } + } + + private static IOException makeIOException(Exception e) { + return new IOException(e); + } + + public String describe() { + return "RabbitIndexWriter\n" + + "\t" + serverHost + ":" + serverPort + " : URL of RabbitMQ server\n" + + "\t" + RabbitMQConstants.SERVER_VIRTUAL_HOST + " : Virtualhost name\n" + + "\t" + RabbitMQConstants.SERVER_USERNAME + " : Username for authentication\n" + + "\t" + RabbitMQConstants.SERVER_PASSWORD + " : Password for authentication\n" + + "\t" + RabbitMQConstants.COMMIT_SIZE + " : Buffer size when sending to RabbitMQ (default 250)\n"; + } } diff --git a/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMQConstants.java b/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMQConstants.java index df4722e5b..86d88d01d 100644 --- a/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMQConstants.java +++ b/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMQConstants.java @@ -17,28 +17,28 @@ package org.apache.nutch.indexwriter.rabbit; interface RabbitMQConstants { - String RABBIT_PREFIX = "rabbitmq.indexer"; + String RABBIT_PREFIX = "rabbitmq.indexer"; - String SERVER_HOST = RABBIT_PREFIX + "server.host"; + String SERVER_HOST = RABBIT_PREFIX + "server.host"; - String SERVER_PORT = RABBIT_PREFIX + "server.port"; + String SERVER_PORT = RABBIT_PREFIX + "server.port"; - String SERVER_VIRTUAL_HOST = RABBIT_PREFIX + "server.virtualhost"; + String SERVER_VIRTUAL_HOST = RABBIT_PREFIX + "server.virtualhost"; - String SERVER_USERNAME = RABBIT_PREFIX + "server.username"; + String SERVER_USERNAME = RABBIT_PREFIX + "server.username"; - String SERVER_PASSWORD = RABBIT_PREFIX + "server.password"; + String SERVER_PASSWORD = RABBIT_PREFIX + "server.password"; - String EXCHANGE_SERVER = RABBIT_PREFIX + "exchange.server"; + String EXCHANGE_SERVER = RABBIT_PREFIX + "exchange.server"; - String EXCHANGE_TYPE = RABBIT_PREFIX + "exchange.type"; + String EXCHANGE_TYPE = RABBIT_PREFIX + "exchange.type"; - String QUEUE_NAME = RABBIT_PREFIX + "queue.name"; + String QUEUE_NAME = RABBIT_PREFIX + "queue.name"; - String QUEUE_DURABLE = RABBIT_PREFIX + "queue.durable"; + String QUEUE_DURABLE = RABBIT_PREFIX + "queue.durable"; - String QUEUE_ROUTING_KEY = RABBIT_PREFIX + "queue.routingkey"; + String QUEUE_ROUTING_KEY = RABBIT_PREFIX + "queue.routingkey"; - String COMMIT_SIZE = RABBIT_PREFIX + "commit.size"; + String COMMIT_SIZE = RABBIT_PREFIX + "commit.size"; } diff --git a/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMessage.java b/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMessage.java index ec76174f5..73fc6fa02 100644 --- a/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMessage.java +++ b/src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMessage.java @@ -22,50 +22,50 @@ import java.util.List; class RabbitMessage { - private List<RabbitDocument> docsToWrite = new LinkedList<>(); - private List<RabbitDocument> docsToUpdate = new LinkedList<>(); - private List<String> docsToDelete = new LinkedList<>(); + private List<RabbitDocument> docsToWrite = new LinkedList<>(); + private List<RabbitDocument> docsToUpdate = new LinkedList<>(); + private List<String> docsToDelete = new LinkedList<>(); - boolean addDocToWrite (RabbitDocument doc) { - return docsToWrite.add(doc); - } + boolean addDocToWrite (RabbitDocument doc) { + return docsToWrite.add(doc); + } - boolean addDocToUpdate (RabbitDocument doc) { - return docsToUpdate.add(doc); - } + boolean addDocToUpdate (RabbitDocument doc) { + return docsToUpdate.add(doc); + } - boolean addDocToDelete (String url) { - return docsToDelete.add(url); - } + boolean addDocToDelete (String url) { + return docsToDelete.add(url); + } - byte[] getBytes() { - Gson gson = new Gson(); - return gson.toJson(this).getBytes(); - } + byte[] getBytes() { + Gson gson = new Gson(); + return gson.toJson(this).getBytes(); + } - boolean isEmpty () { - return docsToWrite.isEmpty() && docsToUpdate.isEmpty() && docsToDelete.isEmpty(); - } + boolean isEmpty () { + return docsToWrite.isEmpty() && docsToUpdate.isEmpty() && docsToDelete.isEmpty(); + } - public List<RabbitDocument> getDocsToWrite() { - return docsToWrite; - } + public List<RabbitDocument> getDocsToWrite() { + return docsToWrite; + } - public List<RabbitDocument> getDocsToUpdate() { - return docsToUpdate; - } + public List<RabbitDocument> getDocsToUpdate() { + return docsToUpdate; + } - public List<String> getDocsToDelete() { - return docsToDelete; - } + public List<String> getDocsToDelete() { + return docsToDelete; + } - public int size () { - return docsToWrite.size() + docsToUpdate.size() + docsToDelete.size(); - } + public int size () { + return docsToWrite.size() + docsToUpdate.size() + docsToDelete.size(); + } - public void clear() { - docsToWrite.clear(); - docsToUpdate.clear(); - docsToDelete.clear(); - } + public void clear() { + docsToWrite.clear(); + docsToUpdate.clear(); + docsToDelete.clear(); + } + docsToWrite.clear(); +

docsToUpdate.clear(); + docsToDelete.clear(); + } } diff --git a/src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrIndexWriter.java b/src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrIndexWriter.java index 19b6c1e3f..10000a78e 100644 --- a/src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrIndexWriter.java +++ b/src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrIndexWriter.java @@ -40,13 +40,6 @@ import org.slf4j.Logger; import org.slf4j.LoggerFactory; -import org.apache.hadoop.fs.Path; -import org.apache.hadoop.fs.FileStatus; -import org.apache.hadoop.fs.FileSystem; -import org.apache.nutch.util.HadoopFSUtil; -import org.apache.hadoop.util.StringUtils; -import org.apache.nutch.util.NutchConfiguration; - // WORK AROUND FOR NOT REMOVING URL ENCODED URLS!!! import java.net.URLDecoder; diff --git a/src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrUtils.java b/src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrUtils.java index ef125989e..c8ad54b38 100644 --- a/src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrUtils.java +++ b/src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrUtils.java @@ -18,7 +18,6 @@ import java.lang.invoke.MethodHandles; import java.util.ArrayList; -import java.util.List; import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/HTMLLanguageParser.java b/src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/HTMLLanguageParser.java index efc821ab3..68f1b6fac 100644 --- a/src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/HTMLLanguageParser.java +++ b/src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/HTMLLanguageParser.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.analysis.lang; -// JDK imports import java.lang.invoke.MethodHandles; import java.util.Enumeration; import java.util.HashMap; diff --git a/src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/LanguageIndexingFilter.java b/src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/LanguageIndexingFilter.java index cd954c70d..6336afa78 100644 --- a/src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/LanguageIndexingFilter.java +++ b/src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/LanguageIndexingFilter.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.analysis.lang; -// Nutch imports import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.crawl.Inlinks; import org.apache.nutch.indexer.IndexingFilter; @@ -30,7 +29,6 @@ import java.util.HashSet; import java.util.Set; -// Hadoop imports import org.apache.hadoop.conf.Configuration; /** diff --git a/src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpBase.java b/src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpBase.java index d218cbc98..d9284c9aa 100644 --- a/src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpBase.java +++ b/src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpBase.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.protocol.http.api; -// JDK imports import java.lang.invoke.MethodHandles; import java.io.BufferedReader; import java.io.IOException; @@ -30,11 +29,9 @@ import java.util.Set; import java.util.concurrent.ThreadLocalRandom; -// Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Nutch imports import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.metadata.Nutch; import org.apache.nutch.net.protocols.Response; @@ -47,12 +44,10 @@ import org.apache.nutch.util.DeflateUtils; import org.apache.hadoop.util.StringUtils; -// Hadoop imports import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; -// crawler-commons imports import crawlercommons.robots.BaseRobotRules; public abstract class HttpBase implements Protocol { diff --git a/src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpException.java b/src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpException.java index ff7ef5b3c..d7ee51a94 100644 --- a/src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpException.java +++ b/src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpException.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.protocol.http.api; -// Nutch imports import org.apache.nutch.protocol.ProtocolException; public class HttpException extends ProtocolException { diff --git a/src/plugin/lib-regex-filter/src/java/org/apache/nutch/urlfilter/api/RegexURLFilterBase.java b/src/plugin/lib-regex-filter/src/java/org/apache/nutch/urlfilter/api/RegexURLFilterBase.java index 154f9e1a1..ecbe29d6f 100644 --- a/src/plugin/lib-regex-filter/src/java/org/apache/nutch/urlfilter/api/RegexURLFilterBase.java +++ b/src/plugin/lib-regex-filter/src/java/org/apache/nutch/urlfilter/api/RegexURLFilterBase.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.urlfilter.api; -// JDK imports import java.lang.invoke.MethodHandles; import java.io.File; import java.io.Reader; @@ -29,15 +28,11 @@ import java.util.List; import java.util.ArrayList; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Hadoop imports import org.apache.hadoop.conf.Configuration; - -// Nutch imports -import org.apache.nutch.net.*; +import org.apache.nutch.net.URLFilter; import org.apache.nutch.util.URLUtil; /** diff --git a/src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagIndexingFilter.java b/src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagIndexingFilter.java index e50a15046..8fbfb582e 100644 --- a/src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagIndexingFilter.java +++ b/src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagIndexingFilter.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.microformats.reltag; -// Nutch imports import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.crawl.Inlinks; import org.apache.nutch.indexer.IndexingFilter; @@ -25,7 +24,6 @@ import java.util.HashSet; import java.util.Set; -// Hadoop imports import org.apache.hadoop.conf.Configuration; /**

org.apache.hadoop.io.Text; import org.apache.nutch.parse.Parse; -// Hadoop imports import org.apache.hadoop.conf.Configuration; /** diff --git a/src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagParser.java b/src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagParser.java index a4ebc8aea..cc5fc4589 100644 --- a/src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagParser.java +++ b/src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagParser.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.microformats.reltag; -// JDK imports import java.lang.invoke.MethodHandles; import java.net.URL; import java.net.URLDecoder; @@ -28,11 +27,9 @@ import org.w3c.dom.Node; import org.w3c.dom.NodeList; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Nutch imports import org.apache.nutch.metadata.Metadata; import org.apache.nutch.parse.HTMLMetaTags; import org.apache.nutch.parse.Parse; @@ -41,7 +38,6 @@ import org.apache.nutch.protocol.Content; import org.apache.nutch.util.StringUtil; -// Hadoop imports import org.apache.hadoop.conf.Configuration; /** diff --git a/src/plugin/mimetype-filter/src/java/org/apache/nutch/indexer/filter/MimeTypeIndexingFilter.java b/src/plugin/mimetype-filter/src/java/org/apache/nutch/indexer/filter/MimeTypeIndexingFilter.java index e374096cb..99c59a62c 100644 --- a/src/plugin/mimetype-filter/src/java/org/apache/nutch/indexer/filter/MimeTypeIndexingFilter.java +++ b/src/plugin/mimetype-filter/src/java/org/apache/nutch/indexer/filter/MimeTypeIndexingFilter.java @@ -29,7 +29,6 @@ import org.apache.commons.cli.GnuParser; import org.apache.commons.cli.UnrecognizedOptionException; -// Nutch imports import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.util.StringUtils; import org.apache.hadoop.io.Text; diff --git a/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/DOMContentUtils.java b/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/DOMContentUtils.java index 1f1061d39..731003c88 100644 --- a/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/DOMContentUtils.java +++ b/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/DOMContentUtils.java @@ -22,17 +22,17 @@ import java.util.Collection; import java.util.ArrayList; import java.util.HashMap; -import java.util.Stack; import org.apache.nutch.parse.Outlink; import org.apache.nutch.util.NodeWalker; import org.apache.nutch.util.URLUtil; +import org.w3c.dom.NamedNodeMap; +import org.w3c.dom.Node; +import org.w3c.dom.NodeList; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.io.MapWritable; import org.apache.hadoop.io.Text; -import org.w3c.dom.*; - /** * A collection of methods for extracting content from DOM trees. * diff --git a/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HTMLMetaProcessor.java b/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HTMLMetaProcessor.java index 159aa760a..1dae99db6 100644 --- a/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HTMLMetaProcessor.java +++ b/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HTMLMetaProcessor.java @@ -20,7 +20,9 @@ import java.net.URL; import org.apache.nutch.parse.HTMLMetaTags; -import org.w3c.dom.*; +import org.w3c.dom.NamedNodeMap; +import org.w3c.dom.Node; +import org.w3c.dom.NodeList; /** * Class for parsing META Directives from DOM trees. This class handles diff --git a/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HtmlParser.java b/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HtmlParser.java index 9ed9fa4ee..78cd257e4 100644 --- a/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HtmlParser.java +++ b/src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HtmlParser.java @@ -17,28 +17,43 @@ package org.apache.nutch.parse.html; +import java.io.ByteArrayInputStream; +import java.io.DataInputStream; +import java.io.File; +import java.io.FileInputStream; +import java.io.IOException; import java.lang.invoke.MethodHandles; import java.util.ArrayList; import java.util.Map; +import java.util.regex.Matcher; +import java.util.regex.Pattern; import java.net.URL; +import java.net.MalformedURLException; import java.nio.charset.StandardCharsets; -import java.io.*; -import java.util.regex.*; -import org.cyberneko.html.parsers.*; import org.xml.sax.InputSource; import org.xml.sax.SAXException; -import org.w3c.dom.*; -import org.apache.html.dom.*; +import org.w3c.dom.DOMException; +import org.w3c.dom.DocumentFragment; import org.slf4j.Logger; import org.slf4j.LoggerFactory; import org.apache.hadoop.conf.Configuration; +import org.apache.html.dom.HTMLDocumentImpl; import org.apache.nutch.metadata.Metadata; import org.apache.nutch.metadata.Nutch; import org.apache.nutch.parse.HTMLMetaTags; import org.apache.nutch.parse.HtmlParseFilters; import org.apache.nutch.parse.Outlink; import org.apache.nutch.parse.Parse; import org.apache.nutch.parse.ParseData; import org.apache.nutch.parse.ParseImpl; import org.apache.nutch.parse.ParseResult; import org.apache.nutch.parse.ParseStatus; import org.apache.nutch.parse.Parser; import org.apache.nutch.protocol.Content; -import org.apache.hadoop.conf.*; -import org.apache.nutch.parse.*; -import org.apache.nutch.util.*; +import org.apache.nutch.util.EncodingDetector; +import org.apache.nutch.util.NutchConfiguration; +import org.cyberneko.html.parsers.DOMFragmentParser; public class HtmlParser implements Parser { private static final Logger LOG = LoggerFactory @@ -93,13 +108,13 @@ private static String sniffCharacterEncoding(byte[] content) { if (metaMatcher.find()) { Matcher charsetMatcher = charsetPattern.matcher(metaMatcher.group(1)); if (charsetMatcher.find()) - encoding = new String(charsetMatcher.group(1)); + encoding = charsetMatcher.group(1); } if (encoding == null) { // check for HTML5 meta charset metaMatcher = charsetPatternHTML5.matcher(str); if (metaMatcher.find()) { - encoding = new String(metaMatcher.group(1)); + encoding = metaMatcher.group(1); } } if (encoding == null) { @@ -250,7 +265,7 @@ public ParseResult getParse(Content content) { } private DocumentFragment parse(InputSource input)

throws Exception { - if (parserImpl.equalsIgnoreCase("tagsoup")) + if ("tagsoup".equalsIgnoreCase(parserImpl)) return parseTagSoup(input); else return parseNeko(input); @@ -325,7 +340,6 @@ private DocumentFragment parseNeko(InputSource input) throws Exception { } public static void main(String[] args) throws Exception { - // LOG.setLevel(Level.FINE); String name = args[0]; String url = "file:" + name; File file = new File(name); @@ -344,6 +358,7 @@ public static void main(String[] args) throws Exception { } + @Override public void setConf(Configuration conf) { this.conf = conf; this.htmlParseFilters = new HtmlParseFilters(getConf()); @@ -355,6 +370,7 @@ public void setConf(Configuration conf) { Nutch.CACHING_FORBIDDEN_CONTENT); } + @Override public Configuration getConf() { return this.conf; } diff --git a/src/plugin/parse-swf/src/java/org/apache/nutch/parse/swf/SWFParser.java b/src/plugin/parse-swf/src/java/org/apache/nutch/parse/swf/SWFParser.java index a81f4f1f0..1c7d480cb 100644 --- a/src/plugin/parse-swf/src/java/org/apache/nutch/parse/swf/SWFParser.java +++ b/src/plugin/parse-swf/src/java/org/apache/nutch/parse/swf/SWFParser.java @@ -18,24 +18,44 @@ package org.apache.nutch.parse.swf; import java.lang.invoke.MethodHandles; +import java.util.ArrayList; +import java.util.Arrays; +import java.util.HashMap; +import java.util.HashSet; +import java.util.Iterator; +import java.util.Stack; +import java.util.Vector; import java.io.FileInputStream; import java.io.IOException; -import java.util.*; import org.slf4j.Logger; import org.slf4j.LoggerFactory; import org.apache.nutch.metadata.Metadata; import org.apache.nutch.net.protocols.Response; -import org.apache.nutch.parse.*; +import org.apache.nutch.parse.Outlink; +import org.apache.nutch.parse.OutlinkExtractor; +import org.apache.nutch.parse.Parse; +import org.apache.nutch.parse.ParseData; +import org.apache.nutch.parse.ParseImpl; +import org.apache.nutch.parse.ParseResult; +import org.apache.nutch.parse.ParseStatus; +import org.apache.nutch.parse.Parser; import org.apache.nutch.protocol.Content; import org.apache.nutch.util.NutchConfiguration; import org.apache.hadoop.conf.Configuration; -import com.anotherbigidea.flash.interfaces.*; -import com.anotherbigidea.flash.readers.*; -import com.anotherbigidea.flash.structs.*; +import com.anotherbigidea.flash.interfaces.SWFActionBlock; +import com.anotherbigidea.flash.interfaces.SWFActions; +import com.anotherbigidea.flash.interfaces.SWFText; +import com.anotherbigidea.flash.interfaces.SWFVectors; +import com.anotherbigidea.flash.readers.SWFReader; +import com.anotherbigidea.flash.readers.TagParser; +import com.anotherbigidea.flash.structs.AlphaColor; +import com.anotherbigidea.flash.structs.Color; +import com.anotherbigidea.flash.structs.Matrix; +import com.anotherbigidea.flash.structs.Rect; import com.anotherbigidea.flash.writers.SWFActionBlockImpl; import com.anotherbigidea.flash.writers.SWFTagTypesImpl; import com.anotherbigidea.io.InStream; @@ -51,20 +71,24 @@ private Configuration conf = null; public SWFParser() { + //default constructor } + @Override public void setConf(Configuration conf) { this.conf = conf; } + @Override public Configuration getConf() { return conf; } + @Override public ParseResult getParse(Content content) { String text = null; - Vector<Outlink> outlinks = new Vector<Outlink>(); + Vector<Outlink> outlinks = new Vector<>(); try { @@ -163,13 +187,13 @@ public static void main(String[] args) throws IOException { * character codes for the correspnding font glyphs (An empty array denotes a * System Font). */ - protected HashMap<Integer, int[]> fontCodes = new HashMap<Integer, int[]>(); + protected HashMap<Integer, int[]> fontCodes = new HashMap<>(); - public ArrayList<String> strings = new ArrayList<String>(); + public ArrayList<String> strings = new ArrayList<>(); - public HashSet<String> actionStrings = new HashSet<String>(); + public HashSet<String> actionStrings = new HashSet<>(); - public ArrayList<String> urls = new ArrayList<String>(); + public ArrayList<String> urls = new ArrayList<>(); public ExtractText() { super(null); @@ -204,7 +228,7 @@ public String getActionText() { int i = 0; Iterator<String> it = urls.iterator(); while (it.hasNext()) { - res[i] = (String) it.next(); + res[i] = it.next(); i++; } return res; @@ -239,7 +263,6 @@ public SWFVectors tagDefineFont2(int id, int flags, String name, int numGlyphs, int ascent, int descent, int leading, int[] codes, int[] advances, Rect[] bounds, int[] kernCodes1, int[] kernCodes2, int[] kernAdjustments) throws IOException { - // System.out.println("-defineFontInfo id=" + id + ", name=" + name); fontCodes.put(new Integer(id), (codes != null) ? codes : new int[0]); return null; @@ -285,10 +308,12 @@ public SWFText tagDefineText2(int id, Rect bounds, Matrix matrix) protected boolean firstY = true; + @Override public void font(int fontId, int textHeight) { - this.fontId = new Integer(fontId); + this.fontId = fontId; } + @Override public void setY(int y) { if (firstY) firstY = false; @@ -303,8 +328,8 @@ public void setY(int y) { * character, instead they adjust glyphAdvances. We don't handle it at all - * in such cases the text will be all glued together. */ + @Override public void text(int[] glyphIndices, int[] glyphAdvances) { - // System.out.println("-text id=" + fontId); int[] codes = (int[]) fontCodes.get(fontId); if (codes == null) { // unknown font, better not guess @@ -324,45 +349,32 @@ public void text(int[] glyphIndices, int[] glyphAdvances) { } else { chars[i] = (char) (codes[index]); } - // System.out.println("-ch[" + i + "]='" + chars[i] + "'(" + - // (int)chars[i] + ") +" + glyphAdvances[i]); } strings.add(new String(chars)); } + @Override public void color(Color color) { } + @Override public void setX(int x) { } + @Override public void done() { strings.add("\n"); } } + @Override public SWFActions tagDoAction() throws IOException { - // ActionTextWriter actions = new ActionTextWriter(new - // PrintWriter(System.out)); - NutchSWFActions actions = new NutchSWFActions(actionStrings, urls); - return actions; + return new NutchSWFActions(actionStrings, urls); } + @Override public SWFActions tagDoInitAction(int arg0) throws IOException { - // ActionTextWriter actions = new ActionTextWriter(new - // PrintWriter(System.out)); - NutchSWFActions actions = new NutchSWFActions(actionStrings, urls); - return actions; - } - - public void tagGeneratorFont(byte[] arg0) throws IOException { - // TODO Auto-generated method stub - super.tagGeneratorFont(arg0); - } - - public void tagGeneratorText(byte[] arg0) throws IOException { - // TODO

Auto-generated method stub - super.tagGeneratorText(arg0); + return new NutchSWFActions(actionStrings, urls); } } @@ -387,6 +399,7 @@ public NutchSWFActions(HashSet<String> strings, ArrayList<String> urls) { stack = new SmallStack(100, strings); } + @Override public void lookupTable(String[] values) throws IOException { for (int i = 0; i < values.length; i++) { if (!strings.contains(values[i])) @@ -396,17 +409,17 @@ public void lookupTable(String[] values) throws IOException { dict = values; } + @Override public void defineLocal() throws IOException { stack.pop(); super.defineLocal(); } public void getURL(int vars, int mode) { - // System.out.println("-getURL: vars=" + vars + ", mode=" + mode); } + @Override public void getURL(String url, String target) throws IOException { - // System.out.println("-getURL: url=" + url + ", target=" + target); stack.push(url); stack.push(target); strings.remove(url); @@ -416,13 +429,12 @@ public void getURL(String url, String target) throws IOException { } public SWFActionBlock.TryCatchFinally _try(String var) throws IOException { - // stack.push(var); strings.remove(var); return super._try(var); } + @Override public void comment(String var) throws IOException { - // stack.push(var); strings.remove(var); super.comment(var); } diff --git a/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/BoilerpipeExtractorRepository.java b/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/BoilerpipeExtractorRepository.java index 7c0d71bb2..a4146b388 100644 --- a/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/BoilerpipeExtractorRepository.java +++ b/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/BoilerpipeExtractorRepository.java @@ -16,19 +16,15 @@ */ package org.apache.nutch.parse.tika; -import java.lang.ClassLoader; -import java.lang.InstantiationException; import java.util.HashMap; import org.apache.commons.logging.Log; import org.apache.commons.logging.LogFactory; -import org.apache.tika.parser.html.BoilerpipeContentHandler; import de.l3s.boilerpipe.BoilerpipeExtractor; -import de.l3s.boilerpipe.extractors.*; class BoilerpipeExtractorRepository { public static final Log LOG = LogFactory.getLog(BoilerpipeExtractorRepository.class); - public static final HashMap<String, BoilerpipeExtractor> extractorRepository = new HashMap<String, BoilerpipeExtractor>(); + public static final HashMap<String, BoilerpipeExtractor> extractorRepository = new HashMap<>(); /** * Returns an instance of the specified extractor diff --git a/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/HTMLMetaProcessor.java b/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/HTMLMetaProcessor.java index 294bde96f..8d5baec24 100644 --- a/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/HTMLMetaProcessor.java +++ b/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/HTMLMetaProcessor.java @@ -20,7 +20,9 @@ import java.net.URL; import org.apache.nutch.parse.HTMLMetaTags; -import org.w3c.dom.*; +import org.w3c.dom.NamedNodeMap; +import org.w3c.dom.Node; +import org.w3c.dom.NodeList; /** * Class for parsing META Directives from DOM trees. This class handles diff --git a/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/TikaParser.java b/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/TikaParser.java index ea864bec2..8c867d8be 100644 --- a/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/TikaParser.java +++ b/src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/TikaParser.java @@ -52,7 +52,6 @@ import org.slf4j.LoggerFactory; import org.w3c.dom.DocumentFragment; -import org.w3c.dom.Element; import org.xml.sax.ContentHandler; /** diff --git a/src/plugin/parse-zip/src/java/org/apache/nutch/parse/zip/ZipTextExtractor.java b/src/plugin/parse-zip/src/java/org/apache/nutch/parse/zip/ZipTextExtractor.java index cc5336a63..966281daa 100644 --- a/src/plugin/parse-zip/src/java/org/apache/nutch/parse/zip/ZipTextExtractor.java +++ b/src/plugin/parse-zip/src/java/org/apache/nutch/parse/zip/ZipTextExtractor.java @@ -17,7 +17,6 @@ package org.apache.nutch.parse.zip; -// JDK imports import java.lang.invoke.MethodHandles; import java.io.IOException; import java.io.InputStream; @@ -26,14 +25,11 @@ import java.util.zip.ZipInputStream; import java.net.URL; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Hadoop imports import org.apache.hadoop.conf.Configuration; -// Nutch imports import org.apache.nutch.metadata.Metadata; import org.apache.nutch.net.protocols.Response; import org.apache.nutch.parse.Parse; diff --git a/src/plugin/parsefilter-naivebayes/src/java/org/apache/nutch/parsefilter/naivebayes/Classify.java b/src/plugin/parsefilter-naivebayes/src/java/org/apache/nutch/parsefilter/naivebayes/Classify.java index d755ff68d..697b833be 100644 --- a/src/plugin/parsefilter-naivebayes/src/java/org/apache/nutch/parsefilter/naivebayes/Classify.java +++ b/src/plugin/parsefilter-naivebayes/src/java/org/apache/nutch/parsefilter/naivebayes/Classify.java @@ -18,7 +18,6 @@ package org.apache.nutch.parsefilter.naivebayes; import java.io.BufferedReader; -import java.io.FileReader; import java.io.IOException; import java.util.HashMap; import java.io.InputStreamReader; diff --git a/src/plugin/parsefilter-regex/src/java/org/apache/nutch/parsefilter/regex/RegexParseFilter.java b/src/plugin/parsefilter-regex/src/java/org/apache/nutch/parsefilter/regex/RegexParseFilter.java index dff602ff7..2209ceb5f 100644 --- a/src/plugin/parsefilter-regex/src/java/org/apache/nutch/parsefilter/regex/RegexParseFilter.java +++ b/src/plugin/parsefilter-regex/src/java/org/apache/nutch/parsefilter/regex/RegexParseFilter.java @@ -41,8 +41,7 @@ import org.slf4j.Logger; import org.slf4j.LoggerFactory; - -import org.w3c.dom.*; +import org.w3c.dom.DocumentFragment; /** * RegexParseFilter. If a regular expression matches either HTML or @@ -58,9 +57,11 @@ private Configuration conf; private DocumentFragment doc; - private static final Map<String,RegexRule> rules = new HashMap<String,RegexRule>(); + private static final Map<String,RegexRule> rules = new HashMap<>(); - public RegexParseFilter() {} + public RegexParseFilter() { + //default constructor + } public RegexParseFilter(String regexFile) { this.regexFile = regexFile; diff --git a/src/plugin/protocol-file/src/java/org/apache/nutch/protocol/file/FileResponse.java b/src/plugin/protocol-file/src/java/org/apache/nutch/protocol/file/FileResponse.java index b6e74ffdf..4b6666af7 100644 ---

a/src/plugin/protocol-file/src/java/org/apache/nutch/protocol/file/FileResponse.java +++ b/src/plugin/protocol-file/src/java/org/apache/nutch/protocol/file/FileResponse.java @@ -17,12 +17,10 @@ package org.apache.nutch.protocol.file; -// JDK imports import java.net.URL; import java.io.IOException; import java.io.UnsupportedEncodingException; -// Nutch imports import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.protocol.Content; import org.apache.nutch.util.MimeUtil; @@ -30,10 +28,8 @@ import org.apache.nutch.net.protocols.HttpDateFormat; import org.apache.nutch.net.protocols.Response; -// Tika imports import org.apache.tika.Tika; -// Hadoop imports import org.apache.hadoop.conf.Configuration; /********** diff --git a/src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/Client.java b/src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/Client.java index 4939b61bf..f0dc7b84b 100644 --- a/src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/Client.java +++ b/src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/Client.java @@ -27,7 +27,6 @@ import java.net.Socket; import java.util.List; -//import java.util.LinkedList; import org.apache.commons.net.MalformedServerReplyException; diff --git a/src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/FtpRobotRulesParser.java b/src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/FtpRobotRulesParser.java index 7b944e74d..603514bc7 100644 --- a/src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/FtpRobotRulesParser.java +++ b/src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/FtpRobotRulesParser.java @@ -33,7 +33,6 @@ import org.slf4j.LoggerFactory; import crawlercommons.robots.BaseRobotRules; -import crawlercommons.robots.SimpleRobotRules; /** * This class is used for parsing robots for urls belonging to FTP protocol. It diff --git a/src/plugin/protocol-http/src/java/org/apache/nutch/protocol/http/Http.java b/src/plugin/protocol-http/src/java/org/apache/nutch/protocol/http/Http.java index bb4ab61c2..772a6c013 100755 --- a/src/plugin/protocol-http/src/java/org/apache/nutch/protocol/http/Http.java +++ b/src/plugin/protocol-http/src/java/org/apache/nutch/protocol/http/Http.java @@ -16,19 +16,15 @@ */ package org.apache.nutch.protocol.http; -// JDK imports import java.lang.invoke.MethodHandles; import java.io.IOException; import java.net.URL; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Hadoop imports import org.apache.hadoop.conf.Configuration; -// Nutch imports import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.net.protocols.Response; import org.apache.nutch.protocol.ProtocolException; diff --git a/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/Http.java b/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/Http.java index 98a630334..456572b69 100644 --- a/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/Http.java +++ b/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/Http.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.protocol.httpclient; -// JDK imports import java.lang.invoke.MethodHandles; import java.io.InputStream; import java.io.IOException; @@ -36,11 +35,9 @@ import org.w3c.dom.NodeList; import org.w3c.dom.Node; -// Slf4j Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// HTTP Client imports import org.apache.commons.httpclient.Header; import org.apache.commons.httpclient.HostConfiguration; import org.apache.commons.httpclient.HttpClient; @@ -50,11 +47,8 @@ import org.apache.commons.httpclient.params.HttpConnectionManagerParams; import org.apache.commons.httpclient.protocol.Protocol; import org.apache.commons.httpclient.protocol.ProtocolSocketFactory; -// NUTCH-1929 Consider implementing dependency injection for crawl HTTPS sites that use self signed certificates -//import org.apache.commons.httpclient.protocol.SSLProtocolSocketFactory; import org.apache.commons.lang.StringUtils; -// Nutch imports import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.net.protocols.Response; import org.apache.nutch.protocol.ProtocolException; diff --git a/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpAuthenticationFactory.java b/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpAuthenticationFactory.java index 9e064bbb8..c4d0345a6 100644 --- a/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpAuthenticationFactory.java +++ b/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpAuthenticationFactory.java @@ -16,20 +16,16 @@ */ package org.apache.nutch.protocol.httpclient; -// JDK imports import java.lang.invoke.MethodHandles; import java.util.ArrayList; import java.util.Collection; -// Slf4j Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Hadoop imports import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.conf.Configurable; -// Nutch imports import org.apache.nutch.metadata.Metadata; /** diff --git a/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpBasicAuthentication.java b/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpBasicAuthentication.java index 9fd0a10ba..35d6bd553 100644 --- a/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpBasicAuthentication.java +++ b/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpBasicAuthentication.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.protocol.httpclient; -// JDK imports import java.lang.invoke.MethodHandles; import java.util.ArrayList; import java.util.List; @@ -25,14 +24,11 @@ import java.util.regex.Matcher; import java.util.regex.Pattern; -// Commons Codec imports import org.apache.commons.codec.binary.Base64; -// Commons Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; -// Hadoop imports import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.conf.Configurable; diff --git

a/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpResponse.java b/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpResponse.java index 6041e13d9..863a02b37 100644 --- a/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpResponse.java +++ b/src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpResponse.java @@ -16,13 +16,11 @@ */ package org.apache.nutch.protocol.httpclient; -// JDK imports import java.io.ByteArrayOutputStream; import java.io.IOException; import java.io.InputStream; import java.net.URL; -// HTTP Client imports import org.apache.commons.httpclient.Header; import org.apache.commons.httpclient.HttpVersion; import org.apache.commons.httpclient.cookie.CookiePolicy; @@ -32,7 +30,6 @@ import org.apache.commons.httpclient.HttpClient; -// Nutch imports import org.apache.nutch.crawl.CrawlDatum; import org.apache.nutch.metadata.Metadata; import org.apache.nutch.metadata.SpellCheckedMetadata; diff --git a/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/Http.java b/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/Http.java index 14751ff1a..90d2be7bb 100644 --- a/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/Http.java +++ b/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/Http.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.protocol.interactiveselenium; -// JDK imports import java.lang.invoke.MethodHandles; import java.io.IOException; import java.net.URL; diff --git a/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/HttpResponse.java b/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/HttpResponse.java index 5630c05d0..71707de5e 100644 --- a/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/HttpResponse.java +++ b/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/HttpResponse.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.protocol.interactiveselenium; -// JDK imports import java.io.BufferedInputStream; import java.io.EOFException; import java.io.IOException; diff --git a/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/DefaultHandler.java b/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/DefaultHandler.java index 6b6d67bfa..e0d286104 100644 --- a/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/DefaultHandler.java +++ b/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/DefaultHandler.java @@ -20,11 +20,14 @@ import org.openqa.selenium.WebDriver; public class DefaultHandler implements InteractiveSeleniumHandler { - public String processDriver(WebDriver driver) { - return null; - } - public boolean shouldProcessURL(String URL) { - return true; - } + @Override + public String processDriver(WebDriver driver) { + return null; + } + + @Override + public boolean shouldProcessURL(String url) { + return true; + } } diff --git a/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/InteractiveSeleniumHandler.java b/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/InteractiveSeleniumHandler.java index ba6fb34bd..7213c6eed 100644 --- a/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/InteractiveSeleniumHandler.java +++ b/src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/InteractiveSeleniumHandler.java @@ -21,5 +21,5 @@ public interface InteractiveSeleniumHandler { public String processDriver(WebDriver driver); - public boolean shouldProcessURL(String URL); + public boolean shouldProcessURL(String url); } diff --git a/src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/Http.java b/src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/Http.java index 41fd5b1de..ee98af45a 100644 --- a/src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/Http.java +++ b/src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/Http.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.protocol.selenium; -// JDK imports import java.lang.invoke.MethodHandles; import java.io.IOException; import java.net.URL; diff --git a/src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/HttpResponse.java b/src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/HttpResponse.java index 681e838a2..33169e522 100644 --- a/src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/HttpResponse.java +++ b/src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/HttpResponse.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.protocol.selenium; -// JDK imports import java.io.BufferedInputStream; import java.io.EOFException; import java.io.IOException; diff --git a/src/plugin/scoring-opic/src/java/org/apache/nutch/scoring/opic/OPICScoringFilter.java b/src/plugin/scoring-opic/src/java/org/apache/nutch/scoring/opic/OPICScoringFilter.java index dad40de29..530f267f1 100644 --- a/src/plugin/scoring-opic/src/java/org/apache/nutch/scoring/opic/OPICScoringFilter.java +++ b/src/plugin/scoring-opic/src/java/org/apache/nutch/scoring/opic/OPICScoringFilter.java @@ -24,7 +24,6 @@ import java.util.List; import java.util.Map.Entry; -// Slf4j Logging imports import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff -

-git a/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/SimilarityScoringFilter.java b/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/SimilarityScoringFilter.java index 0f905b861..8436b87ab 100644 --- a/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/SimilarityScoringFilter.java +++ b/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/SimilarityScoringFilter.java @@ -17,7 +17,6 @@ package org.apache.nutch.scoring.similarity; import java.util.Collection; -import java.util.List; import java.util.Map.Entry; import org.apache.hadoop.conf.Configuration; diff --git a/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/CosineSimilarity.java b/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/CosineSimilarity.java index c68197006..9c8aeb8d4 100644 --- a/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/CosineSimilarity.java +++ b/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/CosineSimilarity.java @@ -17,7 +17,6 @@ package org.apache.nutch.scoring.similarity.cosine; import java.lang.invoke.MethodHandles; -import java.io.IOException; import java.util.Collection; import java.util.Map.Entry; diff --git a/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/Model.java b/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/Model.java index 5c56dddbf..05b85da88 100644 --- a/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/Model.java +++ b/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/Model.java @@ -19,24 +19,17 @@ import java.lang.invoke.MethodHandles; import java.io.BufferedReader; import java.io.IOException; -import java.io.InputStream; -import java.io.InputStreamReader; import java.util.ArrayList; -import java.util.Arrays; import java.util.HashMap; import java.util.List; import org.apache.hadoop.conf.Configuration; -import org.apache.hadoop.fs.FileStatus; -import org.apache.hadoop.fs.FileSystem; -import org.apache.hadoop.fs.Path; import org.apache.hadoop.util.StringUtils; import org.apache.lucene.analysis.TokenStream; import org.apache.lucene.analysis.tokenattributes.CharTermAttribute; import org.apache.nutch.scoring.similarity.util.LuceneAnalyzerUtil.StemFilterType; import org.apache.nutch.scoring.similarity.util.LuceneTokenizer; import org.apache.nutch.scoring.similarity.util.LuceneTokenizer.TokenizerType; -import org.apache.tika.Tika; import org.slf4j.Logger; import org.slf4j.LoggerFactory; diff --git a/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/util/LuceneAnalyzerUtil.java b/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/util/LuceneAnalyzerUtil.java index c6a1c58ca..7e4c3592e 100644 --- a/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/util/LuceneAnalyzerUtil.java +++ b/src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/util/LuceneAnalyzerUtil.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.scoring.similarity.util; -import java.io.Reader; import java.util.List; import org.apache.lucene.analysis.Analyzer; diff --git a/src/plugin/urlfilter-automaton/src/java/org/apache/nutch/urlfilter/automaton/AutomatonURLFilter.java b/src/plugin/urlfilter-automaton/src/java/org/apache/nutch/urlfilter/automaton/AutomatonURLFilter.java index ae4896d19..39fce4ea9 100644 --- a/src/plugin/urlfilter-automaton/src/java/org/apache/nutch/urlfilter/automaton/AutomatonURLFilter.java +++ b/src/plugin/urlfilter-automaton/src/java/org/apache/nutch/urlfilter/automaton/AutomatonURLFilter.java @@ -16,19 +16,15 @@ */ package org.apache.nutch.urlfilter.automaton; -// JDK imports import java.io.Reader; import java.io.IOException; import java.io.StringReader; import java.util.regex.PatternSyntaxException; -// Hadoop imports import org.apache.hadoop.conf.Configuration; -// Automaton imports import dk.brics.automaton.RegExp; import dk.brics.automaton.RunAutomaton; -import org.apache.nutch.net.*; import org.apache.nutch.urlfilter.api.RegexRule; import org.apache.nutch.urlfilter.api.RegexURLFilterBase; diff --git a/src/plugin/urlfilter-ignoreexempt/src/java/org/apache/nutch/urlfilter/ignoreexempt/ExemptionUrlFilter.java b/src/plugin/urlfilter-ignoreexempt/src/java/org/apache/nutch/urlfilter/ignoreexempt/ExemptionUrlFilter.java index 40128c8ec..07523fe1b 100644 --- a/src/plugin/urlfilter-ignoreexempt/src/java/org/apache/nutch/urlfilter/ignoreexempt/ExemptionUrlFilter.java +++ b/src/plugin/urlfilter-ignoreexempt/src/java/org/apache/nutch/urlfilter/ignoreexempt/ExemptionUrlFilter.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.urlfilter.ignoreexempt; -import org.apache.commons.io.IOUtils; import org.apache.hadoop.conf.Configuration; import org.apache.nutch.net.URLExemptionFilter; import org.apache.nutch.util.NutchConfiguration; @@ -26,12 +25,9 @@ import java.lang.invoke.MethodHandles; import java.io.IOException; -import java.io.InputStream; import java.io.Reader; -import java.util.Arrays; import java.util.regex.Pattern; import java.util.List; -import java.util.ArrayList; /** diff --git a/src/plugin/urlfilter-prefix/src/java/org/apache/nutch/urlfilter/prefix/PrefixURLFilter.java b/src/plugin/urlfilter-prefix/src/java/org/apache/nutch/urlfilter/prefix/PrefixURLFilter.java index 1e85dc376..fe04d842d 100644 --- a/src/plugin/urlfilter-prefix/src/java/org/apache/nutch/urlfilter/prefix/PrefixURLFilter.java +++ b/src/plugin/urlfilter-prefix/src/java/org/apache/nutch/urlfilter/prefix/PrefixURLFilter.java @@ -20,17 +20,15 @@ import org.slf4j.LoggerFactory; import org.apache.hadoop.conf.Configuration; -import org.apache.nutch.net.*; import org.apache.nutch.util.PrefixStringMatcher; import org.apache.nutch.util.TrieStringMatcher; - +import org.apache.nutch.net.URLFilter; import org.apache.nutch.plugin.Extension; import org.apache.nutch.plugin.PluginRepository; import java.lang.invoke.MethodHandles; import java.io.Reader; -import java.io.FileReader; import java.io.BufferedReader; import java.io.InputStreamReader; import java.io.IOException; @@ -79,7 +77,7 @@ public String filter(String url) { private TrieStringMatcher readConfiguration(Reader reader) throws IOException { BufferedReader in = new BufferedReader(reader); - List<String> urlprefixes = new ArrayList<String>(); + List<String> urlprefixes = new ArrayList<>(); String line; while ((line = in.readLine()) !=

null) { diff --git a/src/plugin/urlfilter-regex/src/java/org/apache/nutch/urlfilter/regex/RegexURLFilter.java b/src/plugin/urlfilter-regex/src/java/org/apache/nutch/urlfilter/regex/RegexURLFilter.java index 2988114f0..118cd900c 100644 --- a/src/plugin/urlfilter-regex/src/java/org/apache/nutch/urlfilter/regex/RegexURLFilter.java +++ b/src/plugin/urlfilter-regex/src/java/org/apache/nutch/urlfilter/regex/RegexURLFilter.java @@ -16,7 +16,6 @@ */ package org.apache.nutch.urlfilter.regex; -// JDK imports import java.io.IOException; import java.io.Reader; import java.io.StringReader; diff --git a/src/plugin/urlfilter-suffix/src/java/org/apache/nutch/urlfilter/suffix/SuffixURLFilter.java b/src/plugin/urlfilter-suffix/src/java/org/apache/nutch/urlfilter/suffix/SuffixURLFilter.java index 83e68be2a..53e5ce65b 100644 --- a/src/plugin/urlfilter-suffix/src/java/org/apache/nutch/urlfilter/suffix/SuffixURLFilter.java +++ b/src/plugin/urlfilter-suffix/src/java/org/apache/nutch/urlfilter/suffix/SuffixURLFilter.java @@ -18,11 +18,10 @@ package org.apache.nutch.urlfilter.suffix; import org.apache.hadoop.conf.Configuration; -import org.apache.nutch.net.*; import org.apache.nutch.util.NutchConfiguration; import org.apache.nutch.util.SuffixStringMatcher; - +import org.apache.nutch.net.URLFilter; import org.apache.nutch.plugin.Extension; import org.apache.nutch.plugin.PluginRepository; diff --git a/src/plugin/urlnormalizer-ajax/src/java/org/apache/nutch/net/urlnormalizer/ajax/AjaxURLNormalizer.java b/src/plugin/urlnormalizer-ajax/src/java/org/apache/nutch/net/urlnormalizer/ajax/AjaxURLNormalizer.java index be98ebe3d..36794262f 100644 --- a/src/plugin/urlnormalizer-ajax/src/java/org/apache/nutch/net/urlnormalizer/ajax/AjaxURLNormalizer.java +++ b/src/plugin/urlnormalizer-ajax/src/java/org/apache/nutch/net/urlnormalizer/ajax/AjaxURLNormalizer.java @@ -19,8 +19,6 @@ import java.lang.invoke.MethodHandles; import java.net.URL; -import java.net.URI; -import java.net.URLEncoder; import java.net.URLDecoder; import java.net.MalformedURLException; import java.nio.charset.Charset; diff --git a/src/plugin/urlnormalizer-host/src/java/org/apache/nutch/net/urlnormalizer/host/HostURLNormalizer.java b/src/plugin/urlnormalizer-host/src/java/org/apache/nutch/net/urlnormalizer/host/HostURLNormalizer.java index 86f58e4ac..86fea1bcc 100644 --- a/src/plugin/urlnormalizer-host/src/java/org/apache/nutch/net/urlnormalizer/host/HostURLNormalizer.java +++ b/src/plugin/urlnormalizer-host/src/java/org/apache/nutch/net/urlnormalizer/host/HostURLNormalizer.java @@ -33,7 +33,6 @@ import org.apache.nutch.net.URLNormalizer; import org.apache.nutch.plugin.Extension; import org.apache.nutch.plugin.PluginRepository; -import org.apache.nutch.util.URLUtil; /** * URL normalizer for mapping hosts to their desired form. It takes a simple diff --git a/src/plugin/urlnormalizer-protocol/src/java/org/apache/nutch/net/urlnormalizer/protocol/ProtocolURLNormalizer.java b/src/plugin/urlnormalizer-protocol/src/java/org/apache/nutch/net/urlnormalizer/protocol/ProtocolURLNormalizer.java index 413d78938..e72b0d2f2 100644 --- a/src/plugin/urlnormalizer-protocol/src/java/org/apache/nutch/net/urlnormalizer/protocol/ProtocolURLNormalizer.java +++ b/src/plugin/urlnormalizer-protocol/src/java/org/apache/nutch/net/urlnormalizer/protocol/ProtocolURLNormalizer.java @@ -35,7 +35,6 @@ import org.apache.nutch.net.URLNormalizer; import org.apache.nutch.plugin.Extension; import org.apache.nutch.plugin.PluginRepository; -import org.apache.nutch.util.URLUtil; /** * @author markus@openindex.io diff --git a/src/plugin/urlnormalizer-querystring/src/java/org/apache/nutch/net/urlnormalizer/querystring/QuerystringURLNormalizer.java b/src/plugin/urlnormalizer-querystring/src/java/org/apache/nutch/net/urlnormalizer/querystring/QuerystringURLNormalizer.java index 2e1e6da1d..04f613792 100644 --- a/src/plugin/urlnormalizer-querystring/src/java/org/apache/nutch/net/urlnormalizer/querystring/QuerystringURLNormalizer.java +++ b/src/plugin/urlnormalizer-querystring/src/java/org/apache/nutch/net/urlnormalizer/querystring/QuerystringURLNormalizer.java @@ -19,7 +19,7 @@ import java.lang.invoke.MethodHandles; import java.net.MalformedURLException; import java.net.URL; -import java.util.ArrayList; import java.util.Arrays; import java.util.Collections; import java.util.List; @@ -29,9 +28,6 @@ import org.apache.commons.lang.StringUtils; import org.apache.hadoop.conf.Configuration; import org.apache.nutch.net.URLNormalizer; -import org.apache.nutch.plugin.Extension; -import org.apache.nutch.plugin.PluginRepository; -import org.apache.nutch.util.URLUtil; /** * URL normalizer plugin for normalizing query strings but sorting query string diff --git a/src/plugin/urlnormalizer-slash/src/java/org/apache/nutch/net/urlnormalizer/slash/SlashURLNormalizer.java b/src/plugin/urlnormalizer-slash/src/java/org/apache/nutch/net/urlnormalizer/slash/SlashURLNormalizer.java index 34a8b94dd..ae094aa54 100644 --- a/src/plugin/urlnormalizer-slash/src/java/org/apache/nutch/net/urlnormalizer/slash/SlashURLNormalizer.java +++ b/src/plugin/urlnormalizer-slash/src/java/org/apache/nutch/net/urlnormalizer/slash/SlashURLNormalizer.java @@ -35,7 +35,6 @@ import org.apache.nutch.net.URLNormalizer; import org.apache.nutch.plugin.Extension; import org.apache.nutch.plugin.PluginRepository; -import org.apache.nutch.util.URLUtil; /** * @author markus@openindex.io @@ -57,9 +56,11 @@ // We record a map of hosts and boolean, the boolean denotes whether the host should // have slashes after URL paths. True means slash, false means remove the slash - private static final Map<String,Boolean> slashesMap = new HashMap<String,Boolean>(); + private static final Map<String,Boolean> slashesMap = new HashMap<>(); - public SlashURLNormalizer() {} + public

SlashURLNormalizer() { + //default constructor + } public SlashURLNormalizer(String slashesFile) { this.slashesFile = slashesFile; ---------- This is an automated message from the Apache Git Service. To respond to the message, please log on GitHub and use the URL above to go to the specific comment. For queries about this service, please contact Infrastructure at: users@infra.apache.org

8. SUCCESS: Integrated in Jenkins build Nutch-trunk #3511 (See [https://builds.apache.org/job/Nutch-trunk/3511/]) NUTCH-2516 Hadoop imports use wildcards (lewis.mcgibbney: [https://github.com/apache/nutch/commit/b834b8111742d8648d6789d770b4a66e60433fa0]) * (edit) src/java/org/apache/nutch/fetcher/FetcherThreadEvent.java * (edit) src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/DefaultHandler.java * (edit) src/java/org/apache/nutch/parse/ParsePluginsReader.java * (edit) src/java/org/apache/nutch/scoring/webgraph/NodeReader.java * (edit) src/java/org/apache/nutch/crawl/LinkDbFilter.java * (edit) src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrIndexWriter.java * (edit) src/java/org/apache/nutch/crawl/CrawlDbReader.java * (edit) src/java/org/apache/nutch/tools/FreeGenerator.java * (edit) src/java/org/apache/nutch/tools/CommonCrawlFormatWARC.java * (edit) src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCParseFilter.java * (edit) src/java/org/apache/nutch/fetcher/Fetcher.java * (edit) src/java/org/apache/nutch/hostdb/ReadHostDb.java * (edit) src/java/org/apache/nutch/crawl/LinkDbReader.java * (edit) src/java/org/apache/nutch/scoring/webgraph/WebGraph.java * (edit) src/java/org/apache/nutch/tools/FileDumper.java * (edit) src/java/org/apache/nutch/indexer/IndexerOutputFormat.java * (edit) src/java/org/apache/nutch/net/URLNormalizerChecker.java * (edit) src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpResponse.java * (edit) src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/HttpResponse.java * (edit) src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/util/LuceneAnalyzerUtil.java * (edit) src/plugin/index-replace/src/java/org/apache/nutch/indexer/replace/FieldReplacer.java * (edit) src/plugin/parse-zip/src/java/org/apache/nutch/parse/zip/ZipTextExtractor.java * (edit) src/java/org/apache/nutch/hostdb/HostDatum.java * (edit) src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpBase.java * (edit) src/java/org/apache/nutch/crawl/CrawlDbMerger.java * (edit) src/java/org/apache/nutch/crawl/Inlink.java * (edit) src/java/org/apache/nutch/hostdb/UpdateHostDb.java * (edit) src/java/org/apache/nutch/crawl/CrawlDbReducer.java * (edit) src/java/org/apache/nutch/protocol/Content.java * (edit) src/java/org/apache/nutch/service/resources/ConfigResource.java * (edit) src/plugin/mimetype-filter/src/java/org/apache/nutch/indexer/filter/MimeTypeIndexingFilter.java * (edit) src/plugin/urlfilter-regex/src/java/org/apache/nutch/urlfilter/regex/RegexURLFilter.java * (edit) src/java/org/apache/nutch/service/resources/SeedResource.java * (edit) src/java/org/apache/nutch/parse/ParsePluginList.java * (edit) src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagParser.java * (edit) .gitignore * (edit) src/java/org/apache/nutch/scoring/webgraph/NodeDumper.java * (edit) src/plugin/parsefilter-naivebayes/src/java/org/apache/nutch/parsefilter/naivebayes/Classify.java * (edit) src/java/org/apache/nutch/crawl/URLPartitioner.java * (edit) src/java/org/apache/nutch/service/NutchReader.java * (edit) src/plugin/urlfilter-prefix/src/java/org/apache/nutch/urlfilter/prefix/PrefixURLFilter.java * (edit) src/java/org/apache/nutch/protocol/RobotRulesParser.java * (edit) src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/HttpResponse.java * (edit) src/plugin/protocol-selenium/src/java/org/apache/nutch/protocol/selenium/Http.java * (edit) src/java/org/apache/nutch/net/protocols/Response.java * (edit) src/java/org/apache/nutch/util/GZIPUtils.java * (edit) src/plugin/scoring-opic/src/java/org/apache/nutch/scoring/opic/OPICScoringFilter.java * (edit) src/java/org/apache/nutch/indexer/IndexingJob.java * (edit) src/java/org/apache/nutch/tools/arc/ArcSegmentCreator.java * (edit) src/plugin/parsefilter-regex/src/java/org/apache/nutch/parsefilter/regex/RegexParseFilter.java * (edit) src/java/org/apache/nutch/indexer/CleaningJob.java * (edit) src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HtmlParser.java * (edit) src/java/org/apache/nutch/net/URLFilterChecker.java * (edit) src/plugin/protocol-http/src/java/org/apache/nutch/protocol/http/Http.java * (edit) src/plugin/creativecommons/src/java/org/creativecommons/nutch/CCIndexingFilter.java * (edit) src/java/org/apache/nutch/scoring/webgraph/LinkDumper.java * (edit) src/java/org/apache/nutch/indexer/IndexingFiltersChecker.java * (edit) src/java/org/apache/nutch/parse/ParseUtil.java * (edit) src/plugin/indexer-elastic-rest/src/java/org/apache/nutch/indexwriter/elasticrest/ElasticRestIndexWriter.java * (edit) src/java/org/apache/nutch/hostdb/UpdateHostDbReducer.java * (edit) src/java/org/apache/nutch/indexer/IndexerMapReduce.java * (edit) src/java/org/apache/nutch/service/impl/NodeReader.java * (edit) src/java/org/apache/nutch/parse/ParseText.java * (edit) src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpBasicAuthentication.java * (edit) src/java/org/apache/nutch/webui/model/NutchConfig.java * (edit)

src/java/org/apache/nutch/crawl/LinkDbMerger.java * (edit) src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMQConstants.java * (edit) src/java/org/apache/nutch/crawl/Inlinks.java * (edit) src/java/org/apache/nutch/net/URLExemptionFilter.java * (edit) src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/FtpRobotRulesParser.java * (edit) src/java/org/apache/nutch/protocol/ProtocolFactory.java * (edit) src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/LanguageIndexingFilter.java * (edit) src/java/org/apache/nutch/crawl/CrawlDbFilter.java * (edit) src/java/org/apache/nutch/tools/DmozParser.java * (edit) src/plugin/parse-html/src/java/org/apache/nutch/parse/html/DOMContentUtils.java * (edit) src/java/org/apache/nutch/tools/warc/WARCExporter.java * (edit) src/plugin/urlnormalizer-host/src/java/org/apache/nutch/net/urlnormalizer/host/HostURLNormalizer.java * (edit) src/java/org/apache/nutch/segment/SegmentReader.java * (edit) src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/Http.java * (edit) src/java/org/apache/nutch/util/URLUtil.java * (edit) src/java/org/apache/nutch/util/DeflateUtils.java * (edit) src/plugin/protocol-file/src/java/org/apache/nutch/protocol/file/FileResponse.java * (edit) src/java/org/apache/nutch/parse/Parser.java * (edit) src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/Model.java * (edit) src/java/org/apache/nutch/parse/ParseImpl.java * (edit) src/java/org/apache/nutch/scoring/webgraph/LinkRank.java * (edit) src/java/org/apache/nutch/service/impl/NutchServerPoolExecutor.java * (edit) src/java/org/apache/nutch/crawl/CrawlDatum.java * (edit) src/plugin/indexer-elastic/src/java/org/apache/nutch/indexwriter/elastic/ElasticIndexWriter.java * (edit) src/plugin/urlnormalizer-slash/src/java/org/apache/nutch/net/urlnormalizer/slash/SlashURLNormalizer.java * (edit) src/java/org/apache/nutch/crawl/Generator.java * (edit) src/java/org/apache/nutch/net/URLFilter.java * (edit) src/java/org/apache/nutch/fetcher/FetcherOutputFormat.java * (edit) src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/Http.java * (edit) src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitIndexWriter.java * (edit) src/java/org/apache/nutch/parse/HtmlParseFilter.java * (edit) src/plugin/headings/src/java/org/apache/nutch/parse/headings/HeadingsParseFilter.java * (edit) src/plugin/language-identifier/src/java/org/apache/nutch/analysis/lang/HTMLLanguageParser.java * (edit) src/java/org/apache/nutch/crawl/SignatureFactory.java * (edit) src/plugin/urlnormalizer-ajax/src/java/org/apache/nutch/net/urlnormalizer/ajax/AjaxURLNormalizer.java * (edit) src/java/org/apache/nutch/service/model/request/JobConfig.java * (edit) src/java/org/apache/nutch/parse/ParserFactory.java * (edit) src/java/org/apache/nutch/fetcher/QueueFeeder.java * (edit) src/java/org/apache/nutch/protocol/Protocol.java * (edit) src/java/org/apache/nutch/indexer/NutchField.java * (edit) src/plugin/indexer-solr/src/java/org/apache/nutch/indexwriter/solr/SolrUtils.java * (edit) src/plugin/index-more/src/java/org/apache/nutch/indexer/more/MoreIndexingFilter.java * (edit) src/java/org/apache/nutch/fetcher/FetchNodeDb.java * (edit) src/java/org/apache/nutch/tools/AbstractCommonCrawlFormat.java * (edit) src/plugin/urlfilter-automaton/src/java/org/apache/nutch/urlfilter/automaton/AutomatonURLFilter.java * (edit) src/plugin/microformats-reltag/src/java/org/apache/nutch/microformats/reltag/RelTagIndexingFilter.java * (edit) src/plugin/parse-swf/src/java/org/apache/nutch/parse/swf/SWFParser.java * (edit) src/plugin/lib-http/src/java/org/apache/nutch/protocol/http/api/HttpException.java * (edit) src/java/org/apache/nutch/indexer/IndexWriter.java * (edit) src/java/org/apache/nutch/indexer/IndexingFilter.java * (edit) src/plugin/feed/src/java/org/apache/nutch/parse/feed/FeedParser.java * (edit) src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/cosine/CosineSimilarity.java * (edit) src/java/org/apache/nutch/parse/ParseData.java * (edit) src/java/org/apache/nutch/hostdb/UpdateHostDbMapper.java * (edit) src/plugin/urlfilter-ignoreexempt/src/java/org/apache/nutch/urlfilter/ignoreexempt/ExemptionUrlFilter.java * (edit) src/java/org/apache/nutch/tools/WARCUtils.java * (edit) src/plugin/feed/src/java/org/apache/nutch/indexer/feed/FeedIndexingFilter.java * (edit) src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/BoilerpipeExtractorRepository.java * (edit) src/java/org/apache/nutch/fetcher/FetcherThread.java * (edit) src/plugin/protocol-httpclient/src/java/org/apache/nutch/protocol/httpclient/HttpAuthenticationFactory.java * (edit) src/java/org/apache/nutch/util/MimeUtil.java * (edit) src/java/org/apache/nutch/indexer/IndexingFilters.java * (edit) src/java/org/apache/nutch/scoring/webgraph/ScoreUpdater.java * (edit) src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/TikaParser.java * (edit) src/java/org/apache/nutch/metadata/HttpHeaders.java * (edit) src/java/org/apache/nutch/parse/ParseSegment.java * (edit) src/java/org/apache/nutch/indexer/IndexWriters.java * (edit) src/plugin/parse-html/src/java/org/apache/nutch/parse/html/HTMLMetaProcessor.java * (edit) src/java/org/apache/nutch/crawl/CrawlDb.java * (edit) src/plugin/urlnormalizer-querystring/src/java/org/apache/nutch/net/urlnormalizer/querystring/QuerystringURLNormalizer.java * (edit) src/java/org/apache/nutch/tools/CommonCrawlDataDumper.java * (edit) src/plugin/protocol-interactiveselenium/src/java/org/apache/nutch/protocol/interactiveselenium/handlers/InteractiveSeleniumHandler.java * (edit) src/java/org/apache/nutch/crawl/DeduplicationJob.java * (edit) src/plugin/scoring-similarity/src/java/org/apache/nutch/scoring/similarity/SimilarityScoringFilter.java * (edit)

src/java/org/apache/nutch/util/DomUtil.java * (edit) src/java/org/apache/nutch/util/ProtocolStatusStatistics.java * (edit) src/plugin/protocol-ftp/src/java/org/apache/nutch/protocol/ftp/Client.java * (edit) src/java/org/apache/nutch/metadata/CreativeCommons.java * (edit) src/plugin/urlnormalizer-protocol/src/java/org/apache/nutch/net/urlnormalizer/protocol/ProtocolURLNormalizer.java * (edit) src/plugin/lib-regex-filter/src/java/org/apache/nutch/urlfilter/api/RegexURLFilterBase.java * (edit) src/plugin/urlfilter-suffix/src/java/org/apache/nutch/urlfilter/suffix/SuffixURLFilter.java * (edit) src/plugin/parse-tika/src/java/org/apache/nutch/parse/tika/HTMLMetaProcessor.java * (edit) src/java/org/apache/nutch/parse/ParseOutputFormat.java * (edit) src/plugin/indexer-rabbit/src/java/org/apache/nutch/indexwriter/rabbit/RabbitMessage.java NUTCH-2516 Hadoop imports use wildcards (lewis.mcgibbney: [https://github.com/apache/nutch/commit/eff0b862ea5ea8442d0821582f470a830e118e6b]) * (edit) .gitignore NUTCH-2516 Hadoop imports use wildcards (lewis.mcgibbney: [https://github.com/apache/nutch/commit/303fd19c51de7b79cd02cd8dd5583d80050e7e1e]) * (delete) ivy/ivy-2.4.0.jar

9. Bulk close of issues resolved for 1.15