Item 335
git_comments:

1. * 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.
2. * * A vehicle to load a list of Solr documents into a local or remote {@link org.apache.solr.client.solrj.SolrClient}.
3. * * Query solr, and stream the results. Unlike the standard query, this will * send events for each Document rather then add them to the QueryResponse. * * Although this function returns a 'QueryResponse' it should be used with care * since it excludes anything that was passed to callback. Also note that * future version may pass even more info to the callback and may not return * the results in the QueryResponse. * * @since solr 4.0
4. * * Performs an explicit commit, causing pending documents to be committed for indexing * <p> * waitFlush=true and waitSearcher=true to be inline with the defaults for plain HTTP access * @throws IOException If there is a low-level I/O error.
5. * * Performs an explicit optimize, causing a merge of all segments to one. * <p> * Note: In most cases it is not required to do explicit optimize * @param waitFlush block until index changes are flushed to disk * @param waitSearcher block until a new searcher is opened and registered as the main query searcher, making the changes visible * @param maxSegments optimizes down to at most this number of segments * @throws IOException If there is a low-level I/O error.
6. * * Performs an explicit optimize, causing a merge of all segments to one. * <p> * Note: In most cases it is not required to do explicit optimize * @param waitFlush block until index changes are flushed to disk * @param waitSearcher block until a new searcher is opened and registered as the main query searcher, making the changes visible * @throws IOException If there is a low-level I/O error.
7. * * Performs an explicit optimize, causing a merge of all segments to one. * <p> * waitFlush=true and waitSearcher=true to be inline with the defaults for plain HTTP access * <p> * Note: In most cases it is not required to do explicit optimize * @throws IOException If there is a low-level I/O error.
8. * 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.
9. * * Performs an explicit commit, causing pending documents to be committed for indexing * @param waitFlush block until index changes are flushed to disk * @param waitSearcher block until a new searcher is opened and registered as the main query searcher, making the changes visible * @throws IOException If there is a low-level I/O error.
10. * * Release allocated resources. * * @since solr 4.0
11. * * Abstraction through which all communication with a Solr server may be routed * * @since 5.0, replaced {@code SolrServer}
12. * * @param zkHost * A zookeeper client endpoint. * @param updatesToLeaders * If true, sends updates only to shard leaders. * @param httpClient * the {@link HttpClient} instance to be used for all requests. The provided httpClient should use a * multi-threaded connection manager. * @see #CloudSolrClient(String) for full description and details on zkHost
13. * * Create a new client object using multiple string values in a Collection * instead of a standard zkHost connection string. Note that this method will * not be used if there is only one String argument - that will use * {@link #CloudSolrClient(String)} instead. * * @param zkHosts * A Java Collection (List, Set, etc) of HOST:PORT strings, one for * each host in the zookeeper ensemble. Note that with certain * Collection types like HashSet, the order of hosts in the final * connect string may not be in the same order you added them. * @param chroot * A chroot value for zookeeper, starting with a forward slash. If no *

chroot is required, use null. * @param httpClient * the {@link HttpClient} instance to be used for all requests. The provided httpClient should use a * multi-threaded connection manager. * @throws IllegalArgumentException * if the chroot value does not start with a forward slash. * @see #CloudSolrClient(String)

14. * 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.
15. * * @param zkHost * A zookeeper client endpoint. * @param lbClient * LBHttpSolrServer instance for requests. * @see #CloudSolrClient(String) for full description and details on zkHost
16. * * @param zkHost * A zookeeper client endpoint. * @param lbClient * LBHttpSolrServer instance for requests. * @param updatesToLeaders * If true, sends updates only to shard leaders. * @see #CloudSolrClient(String) for full description and details on zkHost
17. * * Create a new client object using multiple string values in a Collection * instead of a standard zkHost connection string. Note that this method will * not be used if there is only one String argument - that will use * {@link #CloudSolrClient(String)} instead. * * @param zkHosts * A Java Collection (List, Set, etc) of HOST:PORT strings, one for * each host in the zookeeper ensemble. Note that with certain * Collection types like HashSet, the order of hosts in the final * connect string may not be in the same order you added them. * @param chroot * A chroot value for zookeeper, starting with a forward slash. If no * chroot is required, use null. * @throws IllegalArgumentException * if the chroot value does not start with a forward slash. * @see #CloudSolrClient(String)
18. * * @param zkHost * A zookeeper client endpoint. * @param updatesToLeaders * If true, sends updates only to shard leaders. * @see #CloudSolrClient(String) for full description and details on zkHost
19. * * SolrJ client class to communicate with SolrCloud. * Instances of this class communicate with Zookeeper to discover * Solr endpoints for SolrCloud collections, and then use the * {@link LBHttpSolrClient} to issue requests. * * This class assumes the id field for your documents is called * 'id' - if this is not the case, you must set the right name * with {@link #setIdField(String)}.
20. * 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.
21. * * ConcurrentUpdateSolrClient buffers all added documents and writes * them into open HTTP connections. This class is thread safe. * * Params from {@link UpdateRequest} are converted to http request * parameters. When params change between UpdateRequests a new HTTP * request is started. * * Although any SolrClient request can be made with this implementation, it is * only recommended to use ConcurrentUpdateSolrClient with /update * requests. The class {@link HttpSolrClient} is better suited for the * query interface.
22. * 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.
23. * * LBHttpSolrClient or "LoadBalanced HttpSolrClient" is a load balancing wrapper around * {@link HttpSolrClient}. This is useful when you * have multiple Solr servers and the requests need to be Load Balanced among them. * * Do <b>NOT</b> use this class for indexing in master/slave scenarios since documents must be sent to the * correct master; no inter-node routing is done. * * In SolrCloud (leader/replica) scenarios, it is usually better to use * {@link CloudSolrClient}, but this class may be used * for updates because the server will forward them to the appropriate leader. * * <p/> * It offers automatic

failover when a server goes down and it detects when the server comes back up. * <p/> * Load balancing is done using a simple round-robin on the list of servers. * <p/> * If a request to a server fails by an IOException due to a connection timeout or read timeout then the host is taken * off the list of live servers and moved to a 'dead server list' and the request is resent to the next live server. * This process is continued till it tries all the live servers. If at least one server is alive, the request succeeds, * and if not it fails. * <blockquote><pre> * SolrClient lbHttpSolrClient = new LBHttpSolrClient("http://host1:8080/solr/", "http://host2:8080/solr", "http://host2:8080/solr"); * //or if you wish to pass the HttpClient do as follows * httpClient httpClient = new HttpClient(); * SolrClient lbHttpSolrClient = new LBHttpSolrClient(httpClient, "http://host1:8080/solr/", "http://host2:8080/solr", "http://host2:8080/solr"); * </pre></blockquote> * This detects if a dead server comes alive automatically. The check is done in fixed intervals in a dedicated thread. * This interval can be set using {@link #setAliveCheckInterval} , the default is set to one minute. * <p/> * <b>When to use this?</b> <br/> This can be used as a software load balancer when you do not wish to setup an external * load balancer. Alternatives to this code are to use * a dedicated hardware load balancer or using Apache httpd with mod_proxy_balancer as a load balancer. See <a * href="http://en.wikipedia.org/wiki/Load_balancing_(computing)">Load balancing on Wikipedia</a> * * @since solr 1.4

24. * 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.
25. solrServer = new TestSolrServer(getSolrClient());
26. solrClient = cloudClient; CloudSolrServer doesn't currently support propagating error codes
27. masterClient = createNewSolrClient(masterJetty.getLocalPort());
28. setup the client...
29. setup the client...
30. runs request request.process(client); but we don't have a NamedList response
31. server = concurrentClient = new ConcurrentUpdateSolrServer(addr,32,8);
32. * * @deprecated Use {@link org.apache.solr.client.solrj.impl.HttpSolrClient}
33. XXX concurrent client reports exceptions differently just to be sure the client has sent the doc
34. setup the client...
35. setup the client...

git_commits:

1. **summary:** SOLR-6895: Deprecate SolrServer classes and replace with SolrClient
   message: SOLR-6895: Deprecate SolrServer classes and replace with SolrClient git-svn-id: https://svn.apache.org/repos/asf/lucene/dev/branches/branch_5x@1648706 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Consider renaming SolrServer to SolrClient
   description: This has been niggling at me for a while. Instantiating a new SolrServer object doesn't create a server, it creates a client.

jira_issues_comments:

1. +1, also {{CloudSolrServer}}, {{LBHttpSolrServer}} etc.
2. However EmbeddedSolrServer *does* instantiate a "server", but names matter and would be good to make the appropriate changes here. Maybe now would be a good time to create the newly named classes and deprecate the existing ones for 5x?
3. I think the origins of SolrServer instead of SolrClient go back to here: http://markmail.org/message/yeoqv643eootcscd Essentially modeling the server hence naming it Server. It's never felt great to me though. +1 to change to Client. And since we're already breaking back compat, we should look into other SolrJ cleanups as well. There's probably a ton of code out there that uses SolrJ though - so we should strive to either keep SolrJ4 clients compatible with Solr5, or alternately provide a simple HttpSolrServer wrapper that uses the new SolrClient.
4. I think if we're going over a major version change (4 -> 5), we don't need to worry about binary compatibility here? Anybody upgrading will immediately get compiler errors, and there will be an entry in CHANGES.txt that says "SolrServer is renamed to SolrClient", etc. EmbeddedSolrServer should probably become something like EmbeddedServerSolrClient. Doesn't exactly trip off the tongue, but it describes what it does :-) There are a bunch of other SolrJ changes and cleanups I'd like to get in for 5.0. Mainly, as I'm working on SOLR-6840, I'm trying to get more of our tests to use SolrJ, which exposes shortcomings in the API pretty quickly. I'll create issues as I come across them.
5. bq. I think if we're going over a major version change (4 -> 5), we don't need to worry about binary compatibility here? I'm primarily talking about compatibility with clients which have SolrJ embedded. But we should also consider how to make migration easy for those clients to upgrade their code as well... SolrJ (being the primary Java client) is sort of different than other parts of Solr (where reliance on Java interfaces is always expert level).
6. I'm not sure I follow. If a client has SolrJ embedded, then it's got a specifically versioned JAR file in it somewhere that it's compiled against. At the moment, there aren't any changes to the SolrJ wire protocol, so clients embedding a 4.x jar will still work. If a client wants to upgrade the jar file, then they have to edit their source and recompile, but I think that's expected when going over a major version bump?
7. bq. If a client has SolrJ embedded, then it's got a specifically versioned JAR file in it somewhere that it's compiled against. At the moment, there aren't any changes to the SolrJ wire protocol, so clients embedding a 4.x jar will still work. Correct, this is what I was talking about when I wrote: bq. There's probably a ton of code out there that uses SolrJ though - so we should strive to either keep SolrJ4 clients compatible with Solr5 bq. If a client wants to upgrade the jar file, then they have to edit their source and recompile, but I think that's expected when going over a major version bump? Yes, but we can make that easier or harder. Being a client interface, we should normally still strive for compatibility (in general), and think about how to make everyone's life easier when we do make changes. There's no hard-n-fast rules - we should do what makes the most sense.
8. OK, I understand you now :-) The name changes won't affect the wire protocol, so 4.x clients (so far!) will be able to communicate with 5.0 servers. I'll put together a patch.
9. IMO: if we're making this change, existing classes/interfaces need to be deprecated, not just removed/renamed. Dropping the deprecated stuff can happen in 6.0, methinks. There's a big *unless*... *unless* there's going to be a 4.10.4 release that deprecates such that we can remove/rename in 5.0. Consider someone depending on Solr in an automated build by version, but they aren't going to want to change their indexing code just upgrade to Solr (Server) 5.0, which would bring in SolrJ 5.0 as well into their builds/compilation perhaps.
10. The problem there is that this a rename, not a replacement. I'm not sure how deprecation would work - would we have to duplicate all the classes, and every method everywhere that takes a SolrServer? And if you're depending on Solr core classes for your build, then you're going to have to make code changes going over a major release. That's what major release versions are for!
11. Here's the patch. I kept EmbeddedSolrServer named as it is in the end, as I think it works fine. StreamingSolrServers is renamed to StreamingSolrClients.
12. bq. The problem there is that this a rename, not a replacement. I'm not sure how deprecation would work Create new *Client classes, copied from *Server implementation; gut and deprecate *Server implementation and have it "extend *Client" ones. Would that work for API compatibility of a Solr 4.x indexer using SolrJ 5x?
13. Oh that would work, yes. I'll amend the patch. Thanks!
14. Updated patch, with deprecations after Erik's suggestion. I'd like to commit this today, as it's pretty heavy and will likely get out of sync with trunk pretty quickly.

15. [~romseygeek] did you test that it works? I wonder if somehow a test could be made that ensures that SolrServer stuff still works in a 4x indexer with the renamed stuff. I have no objections, just want to be sure it's working in a backwards compatible way for 4x indexers.
16. bq. Consider renaming SolrServer to SolrClient +1
17. It'll work if you recompile against the new jar. Not sure if you just drop in the new jar to an existing indexer's classpath, I think that probably depends on the JVM and Classloader, but I don't think that's something we can guarantee anyway.
18. Commit 1648697 from [~romseygeek] in branch 'dev/trunk' [https://svn.apache.org/r1648697] SOLR-6895: Deprecate SolrServer classes and replace with SolrClient
19. Commit 1648706 from [~romseygeek] in branch 'dev/branches/branch_5x' [https://svn.apache.org/r1648706] SOLR-6895: Deprecate SolrServer classes and replace with SolrClient
20. Commit 1648710 from [~romseygeek] in branch 'dev/trunk' [https://svn.apache.org/r1648710] SOLR-6895: Remove SolrServer classes from trunk
21. Thanks all!
22. Commit 1648750 from [~romseygeek] in branch 'dev/trunk' [https://svn.apache.org/r1648750] SOLR-6895: Fix rename error in AliasIntegrationTest
23. Commit 1648751 from [~romseygeek] in branch 'dev/branches/branch_5x' [https://svn.apache.org/r1648751] SOLR-6895: Fix rename error in AliasIntegrationTest
24. Commit 1648773 from [~thetaphi] in branch 'dev/trunk' [https://svn.apache.org/r1648773] SOLR-6895: Fix forbidden-apis also in Maven
25. Commit 1648774 from [~thetaphi] in branch 'dev/branches/branch_5x' [https://svn.apache.org/r1648774] Merged revision(s) 1648773 from lucene/dev/trunk: SOLR-6895: Fix forbidden-apis also in Maven