

git_comments:

1. * Licensed to the Apache Software Foundation (ASF) under one * or more * contributor license agreements. 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. For additional information regarding * copyright in this work, please see the NOTICE file in the top level * directory of this distribution.
2. * * Get an entity deleted task * @param collectionScope * @param entityId * @param version * @return
3. * * Get a task for cleaning up latent entity data. If includeVersion = true, the passed version will be cleaned up as well * Otherwise this is a V-1 operation * * @param scope * @param entityId * @param version * @param includeVersion * @return
4. * * Get an entityVersionCreatedTask * @param scope * @param entity * @return
5. * 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.
6. * * Different implementations
7. * * A provider to allow users to configure their queue impl via properties
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. * * Offer the indexoperation message. Some queues may support not returning the future until ack or fail. * Other queues may return the future after ack on the offer. See the implementation documentation for details. * @param operation
10. * * Mark these message as failed. Set the exception in the future on local operation * * @param messages
11. **comment:** * * A temporary interface of our buffer Q to decouple of producer and consumer;
label: code-design
12. * 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.
13. * * Ack all messages so they do not appear again. Meant for transactional queues, and may or may not be implemented. * This will set the future as done in in memory operations * * @param messages
14. * * Perform a take, potentially blocking until up to takesize is available, or timeout has elapsed. * May return less than the take size, but will never return null * * @param takeSize * @param timeout * @param timeUnit * @return A null safe lid
15. we got something, go process it
16. * 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.
17. if we have a future ack it
18. try to add more
19. * * Get the message from our queue
20. nothing to do
21. * * Set our TTL to 1 month. This is high, but in the event of a bug, we want these entries to get removed
22. now see if the key was there
23. * * This is experimental at best. Our SQS size limit is a problem. We shouldn't use this for index operation. Only for * performing
24. look up the values
25. **comment:** pretty print, disabling for speed mapper.enable(SerializationFeature.INDENT_OUTPUT);
label: code-design
26. * 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.
27. stop our timer
28. write to cassandra
29. signal to SQS
30. no op, just let it retry after the queue timeout
31. no op
32. add all our keys for a single round trip
33. * Read the object from Base64 string.
34. * * The name to put in the map
35. SQS doesn't support more than 10
36. * * The message that subclasses our IndexOperationMessage. holds a pointer to the original message
37. remove it from the map
38. the entry was not present in cassandra, ignore this message. Failure should eventually kick it to a DLQ
39. load them into our response
40. * Write the object to a Base64 string.
41. **comment:** * Hacky, copied from CPEntityManager b/c we can't access it here
label: code-design

42. * * Represent the properties required to build an index request

43. * 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.

44. * * Represent the properties required to build an index request

45. * * DO NOT DELETE! Required for Jackson

46. * 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.

47. now get it back

48. de-index request

49. get the operations out

50. now ack the message

51. wait for it to send to SQS

52. * 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.

53. * * Handle the multiple failure junit trace

54. do this so our test gets marked as ignored. Not pretty, but it works

55. * * Created in an attempt to mark no aws cred tests as ignored. Blocked by this issue * <https://github.com/junit-team/junit/issues/116> * * Until then, simply marks as passed, which is a bit dangerous

56. * 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.

57. * * Represents a set of entities

58. * * Return true if the set is empty * @return

59. * * Get the entity from the result set * @param field, Return the entity with the field * @return

60. * * Get the number of entities in this set * @return

61. If we're the master, register ourselves and move on, if we're not, wait for the master to come up

62. **comment:** install(new MapModule()); TODO, re-enable when index module doesn't depend on queue install(new QueueModule());
label: requirement

63. TODO: can't we just sub in the getEntityRepair method here so for every read of a uniqueEntityField we can verify it is correct?

64. not op, we're not migrating properly to this. Make this an event // This check is for testing purposes and for a test that to be able to dynamically turn // off and on delete previous versions so that it can test clean-up on read. if (System.getProperty(EVENTS_DISABLED, "false").equals("true")) { return; } logger.debug("Handling versionCreated for entity {}:{} v {} " + "scope\n name: {}\n owner: {}\n app: {}", new Object[] { entity.getId().getType(), entity.getId().getUuid(), entity.getVersion(), scope.getName(), scope.getOwner(), scope.getApplication()}); CpEntityManagerFactory cpemf = (CpEntityManagerFactory)jemf; final EntityIndex ei = cpemf.getManagerCache().getEntityIndex(scope); ei.deletePreviousVersions(entity.getId(), entity.getVersion());

65. we can't use our candidate result sets here. The repair won't happen since we now have orphaned documents in our index us the EM so the repair process happens

66. @Test(timeout=30000)

67. * * Go around EntityManager and execute query directly against Core Persistence. * Results may include stale index entries.

68. trigger the repair

69. put this into the top of the queue, once it's acked we've been flushed

70. * * Takes the change and reloads an entity with all changes applied in this entity applied. * The resulting entity from calling load will be the previous version of this entity plus * the entity in this object applied to it.

71. * * Audit a unique field, and remove any stale entries in the system * @param field The field to audit within this collection scope. public Observable<Integer> auditUniqueField(final Field field);

72. * * Represents a set of entities

73. * * Return true if the set is empty * @return

74. * * Get the entity from the result set * @param field, Return the entity with the field * @return

75. * * Get the number of entities in this set * @return

76. now loop through and ensure the entities are there.

77. Load a entity for each entityId we retrieved.

78. Short circuit if we don't have any uniqueValues from the given fields.

79. **comment:** bad unique value, delete this, it's inconsistent
label: code-design

80. **comment:** TODO: explore making this an Async process We'll repair it again if we have to
label: code-design

81. else add it to our result set

82. loop through each field, and construct an entity load

83. Get back set of unique values that correspond to collection of fields

84. * * Retrieves all entities that correspond to each field given in the Collection. * @param fields * @return

85. if it's unique, create a function to validate it and add it to the list of concurrent validations use write-first then read strategy

86. * * Command group used for realtime user commands

87. allocate our max size, worst case now get the set of fields back

88. * * Load UniqueValue that matches field from collection or null if that value does not exist. * * @param colScope Collection scope in which to look for field name/value * @param consistencyLevel Consistency level of query * @param fields Field name/value to search for * @return UniqueValueSet containing fields from the collection that exist in cassandra * @throws ConnectionException on error connecting to Cassandra

89. verify no entity returned
90. write an entity with a unique field
91. try to load via the unique field, should have triggered repair
92. user the unique serialization to verify it's been deleted from cassandra
93. do a mark as one test, and a delete as another
94. if we add a second field we get a second entity that is the exact same. Is this expected?
95. use the entity serializationStrategy to remove the entity data.
96. final StringField expectedString = new StringField("yes", "fred", true);
97. newEntity.setField(expectedString);
98. verify the entity is correct. loadReturned);
99. * * Get the currently configured ReadCL that is more consistent than getReadCL * @return
100. * * Get a gauge and create it * @param clazz * @param name * @param gauge * @return
101. * * * @param metricClass * @param klass * @param name * @return
102. * * Get the values for all the keys. If a value does not exist, it won't be present in the map * @param keys * @return
103. * * Get strings from the map * @param keys * @return
104. add it to the entry
105. * * Get multiple values, using the string builder * @param scope * @param keys * @param builder * @param <T> * @return
106. * * Callbacks for performing row operations
107. * * Write the key * @param keysMutation
108. nothing to return
109. now get all columns, including the "old row key value"
110. * * Build the results from the row keys * @param <T>
111. /* * Create a delete method that deletes by Id. This will delete all documents from ES with the same entity Id, * effectively removing all versions of an entity from all index scopes * @param entityId The entityId to remove */ public Future deleteAllVersionsOfEntity(final Id entityId); /* * Takes all the previous versions of the current entity and deletes all previous versions * @param id The id to remove * @param version The max version to retain */ public Future deletePreviousVersions(final Id id, final UUID version);
112. * * Start the consumer
113. * * Stop the consumers
114. * * Amount of time in milliseconds to wait when ES rejects our request before retrying. Provides simple * backpressure
115. * * The queue implementation to use. Values come from <class>QueueProvider.Implementations</class>
116. * * Amount of time to wait when reading from the queue in milliseconds
117. * * The number of worker threads to consume from the queue
118. * * Amount of time to wait when reading from the queue
119. give us 60 seconds to process the message
120. if this has been serialized, it could be null. don't NPE if it is, there's nothing to ack
121. * * No-op, just disregard it
122. **comment:** TODO T.N., this shouldn't live here. This should live at the UG core tier. However the RM/EM are an absolute mess, so until they're refactored, this is it's home
label: code-design
123. * * Get our index info from ES, but clear our cache first * @param aliasType * @return
124. only wait 2 seconds max
125. * * Loop through and start the workers
126. the actively running subscription
127. send the request off to ES
128. stop consuming
129. ack after we process
130. wire up the gauge of inflight messages
131. start in the background
132. collection all the operations into a single stream
133. take since we're in flight
134. name our thread so it's easy to see
135. * * Stop the workers
136. release so we know we've done processing
137. * * clean up cache
138. **comment:** TODO T.N. Remove this when we move the cursor mapping back to core
label: code-design
139. * * Create the SQS client for the specified settings
140. * * Get the region * @return
141. **comment:** pretty print, disabling for speed mapper.enable(SerializationFeature.INDENT_OUTPUT);
label: code-design
142. **comment:** the amazon client is not thread safe, we need to create one per queue
label: requirement
143. no op, swallow
144. * * Mark tests as ignored if now AWS creds are present
145. EntityRef ref = em.getAlias(getEntityType(), name);
146. usergrid-2389: User defined limit in the query is ignored. Fixed it by following usergrid-2389: User defined limit in the query is ignored. Fixed it by adding

git commits:

1. **summary:** Merge branch 'two-dot-o' into USERGRID-405
message: Merge branch 'two-dot-o' into USERGRID-405 Conflicts: stack/core/src/main/java/org/apache/usergrid/corepersistence/CoreModule.java
stack/core/src/main/java/org/apache/usergrid/corepersistence/events/EntityVersionDeletedHandler.java
stack/corepersistence/collection/src/main/java/org/apache/usergrid/persistence/collection/EntityCollectionManager.java
stack/corepersistence/collection/src/main/java/org/apache/usergrid/persistence/collection/guice/CollectionModule.java
stack/corepersistence/collection/src/main/java/org/apache/usergrid/persistence/collection/impl/EntityCollectionManagerFactoryImpl.java
stack/corepersistence/collection/src/main/java/org/apache/usergrid/persistence/collection/impl/EntityCollectionManagerImpl.java
stack/corepersistence/collection/src/main/java/org/apache/usergrid/persistence/collection/impl/EntityDeletedTask.java
stack/corepersistence/collection/src/main/java/org/apache/usergrid/persistence/collection/impl/EntityVersionCleanupTask.java
stack/corepersistence/collection/src/main/java/org/apache/usergrid/persistence/collection/serialization/UniqueValueSerializationStrategy.java
stack/corepersistence/collection/src/main/java/org/apache/usergrid/persistence/collection/serialization/impl/UniqueValueSerializationStrategyImpl.java

stack/corepersistence/collection/src/main/java/org/apache/usergrid/persistence/collection/util/EntityUtils.java
stack/corepersistence/collection/src/test/java/org/apache/usergrid/persistence/collection/impl/EntityVersionCleanupTaskTest.java
stack/corepersistence/graph/src/main/java/org/apache/usergrid/persistence/graph/impl/GraphManagerImpl.java
stack/corepersistence/map/src/main/java/org/apache/usergrid/persistence/map/MapManager.java
stack/corepersistence/queryindex/src/test/java/org/apache/usergrid/persistence/index/guice/TestIndexModule.java

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Fix hotspot from continuous writes of entities (Shawn)

description: Currently, performing a continuous PUT in 2.0 under heavy load causes a hotspot in our cassandra data. Cause: # Under load, entities can be PUT continuously # Asynchronous cleanups run and delete previous versions # These versions are retained in cassandra for long periods of time. This causes severe row bloating before compaction occurs. Solution: For entity data, we only care about the current max version. We should change this column family to store only the maximum data format. We will need to keep the log of previous versions, so that we can bring ES into a consistent state

jira_issues_comments:

1. I'm going to branch off of 381 since we should only perform this migration once.
2. Current impl branch
3. **body:** We also have to create a unique property ledger. We're going to need to follow the same pattern, however, we can restrict the size of the input, greatly reducing this problem of load since we'll only be storing a single property, not the entire entity.
label: code-design
4. GitHub user tnine opened a pull request: <https://github.com/apache/incubator-usergrid/pull/175> quite ready Usergrid 405 Merge of USERGRID-405, USERGRID-427, and USERGRID-381. Address most of USERGRID-365. You can merge this pull request into a Git repository by running: \$ git pull <https://github.com/apache/incubator-usergrid> USERGRID-405 Alternatively you can review and apply these changes as the patch at: <https://github.com/apache/incubator-usergrid/pull/175.patch> To close this pull request, make a commit to your master/trunk branch with (at least) the following in the commit message: This closes #175 ---- commit ee676e78ad8f1905a169a027eb97aa7b1dc7aa7a Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-01-29T18:46:01Z added new entity object for serialization add new guide interface names fix compile issues in core persistence add array field instead of list field adding proxies to new storage spacing changes fixed id problems remove excess fields Test are passing pushing migrations into core persistence pushing migrations into core persistence moving dependencies to core persistence moving dependencies to core persistence Refactor Application Entity Group fixed null issue with index remove observable commit fa82c6dc00cbfa58d7f12e56bdc757bdc4ae4b0c Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-11T17:51:29Z add logic to continue alias buildout despite issues removing commit 29f24a9716086da70d11f3b5d9839280c5ffc650 Author: Todd Nine <tnine@apigee.com> Date: 2015-02-11T21:18:07Z Fixed index alias cache invalidation commit 4781c4088e282ae1671d590f75fb3546fb8ab944 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-12T00:16:03Z adding condition; not sure this is correct commit 2f8993af9d1bcd09153880b3fe317ed4149d899 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-12T20:15:14Z fix tests commit d439036089ec547648338429e8146073fbf70af0 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-12T21:01:35Z change logging commit 07f6cfedac9d5394c69c6c62355f6830bb37d2f2 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-12T21:02:31Z remove connections commit ba189eaf933a6209810f1e72dde80bfe972f288b Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-12T22:27:55Z merged commit 35b95fd14d1792eb6a51dd9f4a3373d241469fc6 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-13T16:26:26Z fix test commit f1b0c8d864042ccc0481550ae3a48f4fb5da874e Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-13T18:32:01Z adding generic to prevent casting at low levels commit 4657c8df7f150f108dfc77af66efd54c034042d3 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-13T22:04:46Z reimplemented filtering for migrations commit db2585ec2671c4722b97adb88afdc55b54b120cd Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-13T22:28:01Z cp tests passing commit e5224695202fd5533ea027964f7fc0e515f50678 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-13T23:04:59Z tests passing again commit 79ec9787c185b3e1a43aba5f456b5568b5735235 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-16T18:07:55Z adding specific migration types commit dab3ee0f39ca5214c04647953a0783f4fdd692ba Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-16T18:18:04Z test fix commit eace1cdc4396fe80d3da55639ce4e6932132175c Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-16T18:18:23Z Merge branch 'USERGRID-273' of <https://git-wip-us.apache.org/repos/asf/incubator-usergrid> into USERGRID-365 commit ed20408b7bfc4d0006c0e084adca0a7e97da34e Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-16T18:47:01Z Move migrations to common interface commit 59777e5952c31f570273ddc94e5e569bdefad468 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-16T20:31:44Z add routing mapping commit 90950908019f131f5874f635af7ece235bea2ce1 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-16T21:57:32Z merge 365-273 commit 4e143574640e7ec7bb7e288191f24e816862c9d9 Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-16T21:58:01Z Merge branch 'USERGRID-273' of <https://git-wip-us.apache.org/repos/asf/incubator-usergrid> into USERGRID-365 commit 748636873e7ba480949294e5a4b3fb1f7dafdc2c Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-16T22:41:26Z remove writestart when there is no version commit 904c5ddff43813b7eb9a273c46b11346853207db Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-16T22:46:19Z add validation for index resource commit a029041fad4935c41855cab861719da2c1fa7e2f Author: Shawn Feldman <sfeldman@apache.org> Date: 2015-02-17T00:27:31Z merge 276 commit d2371f99b4b64dda52e9ff7991e0f6144859779e Author: Todd Nine <tnine@apigee.com> Date: 2015-02-19T23:04:26Z First pass at refactor and re-implementation for new serialization commit 067e78e303cd35c2c0062221f58fa130ae079455 Author: Todd Nine <tnine@apigee.com> Date: 2015-02-19T23:35:32Z Removed unused synchronous interface Removed entity repair, no longer relevant Unique value serialization needs re-work. commit ffa30962d3203f73ee2253038259af3fa7af2d7 Author: Todd Nine <tnine@apigee.com> Date: 2015-02-20T02:49:22Z WIP overwrite commit 8c581a48d48fba5cd76dd018a40288cf300164ea Author: Todd Nine <tnine@apigee.com> Date: 2015-02-24T00:18:18Z Fixed timestamp on write. Also fixed tests. Left unsupported interface values to support migration. We should remove in a +1 release. commit c2989fc702eeb240956e981de637dca68426a68b Author: Todd Nine <tnine@apigee.com> Date: 2015-02-24T04:04:09Z Renamed unique serializers to be easier to locate Added serializer to create row format necessary for unique ledger commit 9f65a973b0b19d8650bf96ee6b245752d9e3c544 Author: Todd Nine <tnine@apigee.com> Date: 2015-02-24T22:31:25Z Updated tasks and interface for getting unique values commit 49fa8123d09b8afe8ff3496a1ce7019d26e9eacf Author: Todd Nine <tnine@apigee.com> Date: 2015-02-25T01:17:43Z Adds test and verifies api works as expected ----
5. Github user asfgit closed the pull request at: <https://github.com/apache/incubator-usergrid/pull/175>