Item 229
git_comments:

git_commits:

1. **summary:** KAFKA-5772 Improve Util classes (#3370)
   message: KAFKA-5772 Improve Util classes (#3370) Utils with static methods should not be instantiated, hence marking the classes `final` and adding a `private` constructor. Reviewers: Randall Hauch <rhauch@gmail.com>, Jason Gustafson <jason@confluent.io>
   label: code-design

github_issues:

github_issues_comments:

github_pulls:

1. **title:** KAFKA-5772 Improve Util classes
   body: Done for https://issues.apache.org/jira/browse/KAFKA-5772 Utils with static methods should not be instantiated, hence marking the classes `final` and adding a `private` constructor. this is consistent w/ some of the Util classes, such as `ByteUtils`, see: https://github.com/apache/kafka/blob/d345d53e4e5e4f74707e2521aa635b93ba3f1e7b/clients/src/main/java/org/apache/kafka/common/utils/ByteUtils.java#L29-L31

github_pulls_comments:

1. Refer to this link for build results (access rights to CI server needed): https://builds.apache.org/job/kafka-pr-jdk7-scala2.11/5472/ Test PASSed (JDK 7 and Scala 2.11).
2. Refer to this link for build results (access rights to CI server needed): https://builds.apache.org/job/kafka-pr-jdk8-scala2.12/5457/ Test PASSed (JDK 8 and Scala 2.12).
3. @rhauch do you mind taking a look? Or do I need to also have a JIRA for this improvement?
4. @rhauch Thanks Randall :+1: I've created a JIRA and update the tilte of the PR :smiley:
5. @matzew, the Apache JIRA should automatically update with a reference to this PR, but I don't see that happening. Perhaps remove the " -" just after the issue number in the PR title?
6. **body:** Actually, on second thought I'm not sure whether these utility methods are considered part of the public API, and if so whether a KIP would be required. The Apache Kafka project are very strict about ensuring backward compatibility, and technically removing the constructor for these classes breaks backward compatibility. Need advice from one of the committers, such as @hachikuji or @ijuma.
   label: code-design
7. These are class are internal.
8. @ijuma @rhauch thanks for clarification
9. @rhauch is there still interest in merging this? Or should I just close it ?
10. @ijuma any reason not to approve and merge this?
11. retest this please.
12. retest this please
13. One issue with `final` classes is that they're harder to mock (i.e. `PowerMock` is needed). For internal classes, the benefit of making classes `final` is not as pronounced as for public classes (since we can make any changes anyway). Still, happy to merge the PR if the tests pass.
14. @ijuma @rhauch thanks for the feedback. I've now also rebased my branch, based on upstream's trunk!
15. should I close it at this point? Or worth to merge ? On Thu, Sep 14, 2017 at 11:43 PM, Randall Hauch <notifications@github.com> wrote: > *@rhauch* approved this pull request. > > I'm going to approve this on a technical basis: the PR is simple and > minimal. I'll leave to @ijuma <https://github.com/ijuma> to decide > whether it's a desirable change. > > — > You are receiving this because you were mentioned. > Reply to this email directly, view it on GitHub > <https://github.com/apache/kafka/pull/3370#pullrequestreview-62895570>, > or mute the thread > <https://github.com/notifications/unsubscribe-auth/AAJnzqMKSOIpyfNS5L6z1xVfhEnjWRsnks5siZ3mgaJpZM4N-LTL> > . > -- Matthias Wessendorf blog: http://matthiaswessendorf.wordpress.com/ twitter: http://twitter.com/mwessendorf
16. @ijuma should I close it or will it be merged ? :-)
17. Thanks for the patch. LGTM.
18. Thanks :smile_cat:

github_pulls_reviews:

1. Is there a reason why this one is not final?
2. @ijuma there was no particular reason, added `final` keyword here to

jira_issues:

1. **summary:** Improve Util classes
   description: Utils with static methods should not be instantiated, hence we should improve them by marking the classes final and adding a private constructor as well. In addition this is consistent w/ a few of the existing Util classes, such as ByteUtils, see: https://github.com/apache/kafka/blob/d345d53e4e5e4f74707e2521aa635b93ba3f1e7b/clients/src/main/java/org/apache/kafka/common/utils/ByteUtils.java#L29-L31

jira_issues_comments:

1. As requested by [~rhauch], for this PR: https://github.com/apache/kafka/pull/3370 I've created this JIRA
2. matzew commented on issue #3370: KAFKA-5772 Improve Util classes URL: https://github.com/apache/kafka/pull/3370#issuecomment-353571386 @ijuma should I close it or will it be merged ? :-) -- 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. hachikuji commented on issue #3370: KAFKA-5772 Improve Util classes URL: https://github.com/apache/kafka/pull/3370#issuecomment-353657827 Thanks for the patch. LGTM. -- 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. hachikuji closed pull request #3370: KAFKA-5772 Improve Util classes URL: https://github.com/apache/kafka/pull/3370 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/clients/src/main/java/org/apache/kafka/clients/ClientUtils.java b/clients/src/main/java/org/apache/kafka/clients/ClientUtils.java index 2a9b7631f39..fba220c8266 100644 --- a/clients/src/main/java/org/apache/kafka/clients/ClientUtils.java +++ b/clients/src/main/java/org/apache/kafka/clients/ClientUtils.java @@ -35,9 +35,11 @@ import static org.apache.kafka.common.utils.Utils.getHost; import static org.apache.kafka.common.utils.Utils.getPort; -public class ClientUtils { +public final class ClientUtils { private static final Logger log = LoggerFactory.getLogger(ClientUtils.class); + private ClientUtils() {} + public static List<InetSocketAddress> parseAndValidateAddresses(List<String> urls) { List<InetSocketAddress> addresses = new ArrayList<>(); for (String url : urls) {

diff --git a/clients/src/main/java/org/apache/kafka/clients/NetworkClientUtils.java b/clients/src/main/java/org/apache/kafka/clients/NetworkClientUtils.java index c4559a48ca8..94fe288090e 100644 --- a/clients/src/main/java/org/apache/kafka/clients/NetworkClientUtils.java +++ b/clients/src/main/java/org/apache/kafka/clients/NetworkClientUtils.java @@ -26,7 +26,9 @@ /** * Provides additional utilities for {@link NetworkClient} (e.g. to implement blocking behaviour). */ -public class NetworkClientUtils { +public final class NetworkClientUtils { + + private NetworkClientUtils() {} /** * Checks whether the node is currently connected, first calling `client.poll` to ensure that any pending diff --git a/clients/src/main/java/org/apache/kafka/common/requests/RequestUtils.java b/clients/src/main/java/org/apache/kafka/common/requests/RequestUtils.java index 103246cac50..a1c27b725b8 100644 --- a/clients/src/main/java/org/apache/kafka/common/requests/RequestUtils.java +++ b/clients/src/main/java/org/apache/kafka/common/requests/RequestUtils.java @@ -35,7 +35,9 @@ import static org.apache.kafka.common.protocol.CommonFields.RESOURCE_NAME_FILTER; import static org.apache.kafka.common.protocol.CommonFields.RESOURCE_TYPE; -class RequestUtils { +final class RequestUtils { + + private RequestUtils() {} static Resource resourceFromStructFields(Struct struct) { byte resourceType = struct.get(RESOURCE_TYPE); diff --git a/clients/src/main/java/org/apache/kafka/common/security/JaasUtils.java b/clients/src/main/java/org/apache/kafka/common/security/JaasUtils.java index 08f41c3349b..c2e7fbdc74a 100644 --- a/clients/src/main/java/org/apache/kafka/common/security/JaasUtils.java +++ b/clients/src/main/java/org/apache/kafka/common/security/JaasUtils.java @@ -22,7 +22,7 @@ import org.slf4j.Logger; import org.slf4j.LoggerFactory; -public class JaasUtils { +public final class JaasUtils { private static final Logger LOG = LoggerFactory.getLogger(JaasUtils.class); public static final String JAVA_LOGIN_CONFIG_PARAM = "java.security.auth.login.config"; @@ -31,6 +31,8 @@ public static final String ZK_SASL_CLIENT = "zookeeper.sasl.client"; public static final String ZK_LOGIN_CONTEXT_NAME_KEY = "zookeeper.sasl.clientconfig"; + private JaasUtils() {} + public static boolean isZkSecurityEnabled() { boolean zkSaslEnabled = Boolean.parseBoolean(System.getProperty(ZK_SASL_CLIENT, "true")); String zkLoginContextName = System.getProperty(ZK_LOGIN_CONTEXT_NAME_KEY, "Client"); diff --git a/clients/src/main/java/org/apache/kafka/common/security/scram/ScramCredentialUtils.java b/clients/src/main/java/org/apache/kafka/common/security/scram/ScramCredentialUtils.java index 55b0651e8f0..8d80542d2aa 100644 --- a/clients/src/main/java/org/apache/kafka/common/security/scram/ScramCredentialUtils.java +++ b/clients/src/main/java/org/apache/kafka/common/security/scram/ScramCredentialUtils.java @@ -31,12 +31,14 @@ * </pre> * */ -public class ScramCredentialUtils { +public final class ScramCredentialUtils { private static final String SALT = "salt"; private static final String STORED_KEY = "stored_key"; private static final String SERVER_KEY = "server_key"; private static final String ITERATIONS = "iterations"; + private ScramCredentialUtils() {} + public static String credentialToString(ScramCredential credential) { return String.format("%s=%s,%s=%s,%s=%s,%s=%d", SALT, diff --git a/clients/src/main/java/org/apache/kafka/common/utils/CollectionUtils.java b/clients/src/main/java/org/apache/kafka/common/utils/CollectionUtils.java index d7ab4e0b39c..dd4b46a0c63 100644 --- a/clients/src/main/java/org/apache/kafka/common/utils/CollectionUtils.java +++ b/clients/src/main/java/org/apache/kafka/common/utils/CollectionUtils.java @@ -23,7 +23,10 @@ import java.util.List; import java.util.Map; -public class CollectionUtils { +public final class CollectionUtils { + + private CollectionUtils() {} + /** * group data by topic * @param data Data to be partitioned diff --git a/clients/src/main/java/org/apache/kafka/common/utils/Utils.java b/clients/src/main/java/org/apache/kafka/common/utils/Utils.java index 2843f3ef82c..8d8f118981a 100755 --- a/clients/src/main/java/org/apache/kafka/common/utils/Utils.java +++ b/clients/src/main/java/org/apache/kafka/common/utils/Utils.java @@ -59,7 +59,9 @@ import java.util.regex.Matcher; import java.util.regex.Pattern; -public class Utils { +public final class Utils { + + private Utils() {} // This matches URIs of formats: host:port and protocol:\\host:port // IPv6 is supported with [ip] pattern -- 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. matzew commented on issue #3370: KAFKA-5772 Improve Util classes URL: https://github.com/apache/kafka/pull/3370#issuecomment-353667235 Thanks :smile_cat: -- 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