Volta Log Library user manual

Table Of Contents

1 References

[Ref.01] Volta Log distribution package (volta-log-x.y.z.* http://sourceforge.net/projects/voltalog/files/?source=navbar)
[Ref.02] Apache Log4j2 web site (http://logging.apache.org/log4j/2.x/)

2 Volta Log

Volta Log is a java library that enables you to manage the logs of your application, saving them in Cassandra within a designated structure instead of a file. As a result, it ensures faster access to logs and a more efficient management of them.

3 Volta Log license

The license applied to Volta Log is "Apache Software License 2.0" (http://www.apache.org/licenses/LICENSE-2.0.html).

4 System Requirements

To use Volta Log, a JRE (Java Runtime Environment) version 7 or greater (http://www.java.com) is required.

5 How to use Volta Log

The following are the steps needed to use the library Volta Log inside a standalone java project.

5.1 Import libraries

To use the library Volta Log you must import in the java project the following libraries included, for simplicity, in the distribution package of Volta Log [Ref.01]:

- volta-log-x.y.z.jar
- log4j-1.2-api-2.0.2.jar
- log4j-api-2.0.2.jar
- log4j-core-2.0.2.jar
- slf4j-api-1.7.5.jar
- cassandra-driver-core-2.1.0.jar
- guava-16.0.1.jar
- metrics-core-3.0.2.jar
- netty-3.9.0-Final.jar
- snakeyaml-1.11.jar

N.B. xyz indicates a generic version of the library Volta Log.

5.2 Configuring Volta Log as Apache Log4j 2 appender

To use the library Volta Log you must configure the library itself as Apache Log4j 2 appender [Ref.02]; for that purpose put in the java project (eg. the src folder) the file log4j2.xml included, for simplicity, in the data folder of the distribution package Volta Log [Ref.01].

Then you need to edit the above configuration file log4j2.xml, specifying at least the values of the following parameters:

- applicationName: attribute in the xml tag *CassandraLogger* indicating the name of the application that makes the logs; if not specified the default value is *Application*
- server: attribute in the xml tag *Cassandra* indicating the address of one of the Apache Cassandra cluster nodes in which the application makes the logs; this address must be specified
- username: attribute in the xml tag *Cassandra* indicating the username of Apache Cassandra cluster where the application makes the logs; this value must be specified, and you can leave the username *cassandra* of default user
- password: attribute in the xml tag *Cassandra* indicating the password of the Apache Cassandra cluster where the application makes the logs; this value must be specified, and you can leave the password *cassandra* of default user

- keySpace: attribute in the xml tag *Cassandra* indicating the keyspace of Apache Cassandra cluster where the application makes the logs; if not specified the default value is *ApplicationsLOG*; within that keyspace logs table is created that will contain the application logs
- level: attribute in the xml tag *Root* (under tag *Loggers*), indicating the level of the log, that is the the application logs that have a lower priority then this level/threshold will not be produced; the possible values are as follows (from the more permissive to the most restrictive): all, trace, debug, info, warn, error, fatal, off.

For completeness the following tables contain all the parameters related to the xml tags *CassandraLogger* and *Cassandra* in the configuration file log4j2.xml, configurable for use of Volta Log as Apache Log4j 2 appender.

The accepted parameters for the xml tag *CassandraLogger* are the following:

Parameter	Type	Description
name	String	Required. Name of the appender.
applicationName	String	Name of the application that is logging. This value will be used as a partition key for the records related to the current application. If null, the default value is *Application*
ignoreExceptions	boolean	The default value is true, if exceptions occur during the execution of the appender will be recorded internally and then ignored. If set to false, however, exceptions will raise to the caller.
filter	Filter	A Filter determines if the event should be handled by the appender. More Filters can be used using the class CompositeFilter
bufferSize	Int	If greater than 0, the appender enters information into a buffer and writes the log when it reaches the set size.

The accepted parameters for the xml tag *Cassandra* are the following

Parameter	Type	Description
factoryClassName	Class	To make a connection to Cassandra, you can use this attribute and factoryMethodName to specify a class and a method that will return the data connection. The method must return a `voltalog.logger.CassandraProperties`. While using this parameter should not be specified server, port, username and password.
factoryMethodName	Method	See factoryClassName.
keySpace	String	KeySpace on which will be created the log table. If not specified, it will be created a new keyspace *ApplicationsLOG*.
server	String	If you decide not to use the factoryClassName and factoryMethodName, you have to specify the server to which to connect. If not specified, it will use the default *localhost*.
port	Int	If you decide not to use the factoryClassName and factoryMethodName, you must specify the port on which to connect. If not specified, it will use the default *9042*.
username	String	If you decide not to use the factoryClassName and factoryMethodName, you must specify the username to use for the connection.
password	String	If you decide not to use the factoryClassName and factoryMethodName, you must specify the password to use for the connection.

5.3 Using the library to log on Apache Cassandra DB

To log through the library Volta Log, once imported the necessary libraries and configured Volta Log as appender for Apache Log4j 2, you have to use the class org.apache.log4j.Logger [Ref.02].

For example, at any point in the java code of the project, you can enter the following instructions:

```
Logger logger = Logger.getLogger(“Logger name”);

logger.error(“Log message”);
```

the execution of the above sample code will result in the insertion of a tuple in the log table of the appropriate keyspace Apache Cassandra, as configured in the file log4j2.xml. As in the example the log is made with priority *error*, the log will actually be stored in the DB if the *level* specified in the configuration file log4j2.xml is equal to or less restrictive then *error*.