

Cloudera Runtime 7.3.1

## Schema Registry Reference

Date published: 2019-08-22

Date modified: 2024-12-10

# CLOUDERA

<https://docs.cloudera.com/>

# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

**Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.**

# Contents

<b>SchemaRegistryClient properties reference.....</b>	<b>4</b>
<b>KafkaAvroSerializer properties reference.....</b>	<b>6</b>
<b>KafkaAvroDeserializer properties reference.....</b>	<b>7</b>

## SchemaRegistryClient properties reference

Review the following reference for a comprehensive list of the SchemaRegistryClient properties.

Property Name	Description	Type	Default Value
schema.registry.url	The URL of the Schema Registry server which this client connects to.	String	http://localhost:9090/api/v1
schema.registry.client.local.jars.path	The local directory path to which downloaded JARs are copied to.	String	/tmp/schema-registry/local-jars
schema.registry.client.class.loader.cache.size	The maximum size of the classloader cache.	Int	1024
schema.registry.client.class.loader.cache.expiry.interval.in.seconds	The expiry interval (in seconds) of an entry in the classloader cache.	Int	3600 sec
schema.registry.client.schema.version.cache.size	The maximum size of schema text cache.	Int	1024
schema.registry.client.schema.version.cache.expiry.interval.in.seconds	The expiry interval (in seconds) of an entry in schema version cache.	Int	300 sec
schema.registry.client.schema.metadata.cache.size	The maximum size of schema metadata cache.	Int	1024
schema.registry.client.schema.metadata.cache.expiry.interval.in.seconds	The expiry interval (in seconds) of an entry in schema metadata cache.	Int	300 sec
schema.registry.client.schema.text.cache.size	The maximum size of schema text cache.	Int	1024
schema.registry.client.schema.text.cache.expiry.interval.in.seconds	The expiry interval (in seconds) of an entry in schema text cache.	Int	300 sec
schema.registry.client.url.selector	Schema Registry URL selector class.	String	com.hortonworks.registries.schema.registry.client.Fail
sasl.jaas.config	Schema Registry Dynamic JAAS configuration for SASL connection.	String	null
schema.registry.auth.username	Username for basic authentication.	String	null
schema.registry.auth.password	Password for basic authentication.	String	null
schema.registry.hash.function	Hashing algorithm for generating schema fingerprints.	String	MD5
chema.registry.auth.type	The type of authentication the client should use. If the value is oauth2, it will be configured for oauth login, otherwise it will use Kerberos.	String	kerberos
schema.registry.oauth.client.id	Client ID for OAuth server in case of oauth login.	String	empty
schema.registry.oauth.secret	Secret for OAuth server in case of oauth login.	String	empty
schema.registry.oauth.server.url	OAuth server URL in case of oauth login.	String	empty
schema.registry.oauth.scope	OAuth scope in case of oauth login.	String	empty
schema.registry.oauth.request.method	HTTP method for requesting the oauth token	String	post

Property Name	Description	Type	Default Value
connector.provider.class	Classname of a Jersey connector provider. (For example: org.glassfish.jersey.apache.connector.ApacheConnectorProvider) Make sure the class is on classpath. If this is set, Backoff policy might need to be set to com.hortonworks.registries.schemaregistry.retry.policy.ExponentialBackoffPolicy.	String	empty
schema.registry.client.ssl	Schema Registry SSL configuration <sup>1</sup>	Object	null
schema.registry.client.retry.policy	Schema Registry Client retry configuration <sup>2</sup>	Object	ExponentialBackoffPolicy with the following default properties:  <pre>sleepTimeMs=1000 maxAttempts=5 timeoutMs=90000</pre>

### Related Information

[Compatibility policies](#)

- <sup>1</sup> The schema.registry.client.ssl property expects a map of SSL configuration properties. The following snippet lists all valid properties.

```
protocol: SSL
hostnameVerifierClass: com.hortonworks.registries.schemaregistry.avro.util
.AcceptAllHostnameVerifier
keyStoreType: JKS
keyStorePath: ./src/test/resources/jks/client.jks
keyStorePassword: clientpwd
keyPassword:
keyStoreProvider:
keyManagerFactoryProvider:
keyManagerFactoryAlgorithm:
trustStoreType: JKS
trustStorePath: ./src/test/resources/jks/client.jks
trustStorePassword: clientpwd
trustStoreProvider:
trustManagerFactoryProvider:
trustManagerFactoryAlgorithm:
```

- <sup>2</sup> The schema.registry.client.retry.policy expects a retry policy and a number of configuration properties related to the retry policy.

The valid policies are as follows:

```
com.hortonworks.registries.schemaregistry.retry.policy.NOOPBackoffPolicy
com.hortonworks.registries.schemaregistry.retry.policy.FixedTimeBackoffPolicy
com.hortonworks.registries.schemaregistry.retry.policy.ExponentialBackoffPolicy
```

The valid properties of the policies are as follows:

```
sleepTimeMs=[***SLEEPTIME VALUE***]
maxAttempts=[***MAX ATTEMPT VALUE***]
timeoutMs=[***TIMEOUT VALUE***]
```

## KafkaAvroSerializer properties reference

Review the following reference for a comprehensive list of the KafkaAvroSerializer properties.

Property Name	Description	Type	Default Value
schema.compatibility	The default compatibility of the new schemas created by the serializer. For more information about the compatibility options, see the <i>Compatibility policies</i> documentation.	String	BACKWARD
schema.group	The default group of the new schemas created by the serializer.	String	kafka
schema.name.key.suffix	The provided string will be appended to the end of the schema name created for record keys. The schema name is the topic name initially, so as a result, you will have topic name + provided suffix as schema name.	String	:k
schema.name.value.suffix	The provided string will be appended to the end of the schema name created for record values. The schema name is the topic name initially, so as a result, you will have topic name + provided suffix as schema name.	String	null
null.passthrough.enabled	If this is enabled, the serializer will return immediately with null value, without putting anything in the serialized output. Having null values can be important in compact topics. Having null as the latest value for a key indicates for Kafka that the corresponding key can be deleted during log compaction.	Boolean	false
logical.type.conversion.enabled	<p>If this is enabled, Avro record fields can be serialized that are instances of Java classes (BigDecimal, UUID, LocalDate, and so on) that can be serialized in Avro using logical types. For more information, see the <a href="#">Logical Types</a> section in the official Avro documentation.</p> <p>This does not have to be enabled if SpecificRecords are used as part of generated classes. SpecificDatumWriter is used for serializing SpecificRecord instances and that writer handles logical types automatically.</p>	Boolean	false

Property Name	Description	Type	Default Value
serdes.protocol.version	The protocol version to be used. The available default registered protocols are the following: <ul style="list-style-type: none"> <li>CONFLUENT_VERSION_PROTOCOL = 0x0;</li> <li>METADATA_ID_VERSION_PROTOCOL = 0x1;</li> <li>VERSION_ID_AS_LONG_PROTOCOL = 0x2;</li> <li>VERSION_ID_AS_INT_PROTOCOL = 0x3;</li> <li>JSON_PROTOCOL = 0x4;</li> </ul>	Byte	0x3 (VERSION_ID_AS_INT_PROTOCOL)
store.schema.version.id.in.header	If this is set to true, the schema version ID will be stored in the record header instead of the value.	Boolean	false
key_schema_version_id_header_name	If the schema of the record key is stored in the header, the header name can be configured.	String	key.schema.version.id
value_schema_version_id_header_name	If the schema of the record value is stored in the header, the header name can be configured.	String	value.schema.version.id

The `KafkaAvroSerializer` contains an embedded `SchemaRegistryClient`. The configuration for `SchemaRegistryClient` should also be provided for the `KafkaAvroSerializer`. For more information about configurable properties for the *Schema Registry Client properties* reference documentation.

### Related Information

[Schema Registry Client properties reference](#)

## KafkaAvroDeserializer properties reference

Review the following reference for a comprehensive list of the `KafkaAvroDeserializer` properties.

Property Name	Description	Type	Default Value
schemaregistry.reader.schema.versions	This is for setting which version of the schema should be used for which topic. <sup>3</sup>	Map<String, Integer>	Empty map
specific.avro.reader	If this is enabled, <code>SpecificDatumReader</code> will be used for deserialization.	Boolean	false

<sup>3</sup> Example:

```
Map<String, Object> configs = new HashMap<>()
Map<String, Integer> readerVersions = new HashMap<>();
readerVersions.put("clicks", 2);
readerVersions.put("users", 1);
configs.put("schemaregistry.reader.schema.versions", readerVersions);
```

Property Name	Description	Type	Default Value
logical.type.conversion.enabled	If this is enabled, Avro record fields that have logical types can be deserialized directly into their corresponding Java classes (BigDecimal, UUID, LocalDate, and so on). If not enabled, the record field will be deserialized as raw type, without handling the logical type. In this case, probably the same logical type related conversion will be needed to get a valid value. If specific.avro.reader is set to true, this does not have any effect.	Boolean	false
key_schema_version_id_header_name	If the schema of the record key is stored in the header, the header name can be configured.	String	key.schema.version.id
value_schema_version_id_header_name	If the schema of the record value is stored in the header, the header name can be configured.	String	value.schema.version.id

The `KafkaAvroDeserializer` contains an embedded `SchemaRegistryClient`. The configuration for `SchemaRegistryClient` should also be provided for the `KafkaAvroDeserializer`. For more information about configurable properties for the *Schema Registry Client properties* reference documentation.

#### Related Information

[Schema Registry Client properties reference](#)