

Gora

Camel-Gora is an [Apache Camel](#) component that allows you to work with NoSQL databases using the [Apache Gora](#) framework.

Available as of Camel 2.14

Maven users will need to add the following dependency to their pom.xml for this component:

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-gora</artifactId>
  <version>x.x.x</version>
  <!-- use the same version as your Camel core version -->
</dependency>
```

Apache Gora Overview

The [Apache Gora](#) open source framework provides an in-memory data model and persistence for big data. Gora supports persisting to column stores, key value stores, document stores and RDBMSs, and analyzing the data with extensive [Apache Hadoop™ MapReduce](#) support. Gora uses the [Apache Software License v2.0](#) and graduated from the Apache Incubator in January 2012 to become a top-level Apache project.

Apache Gora currently supports the following datastores: [Apache HBase](#), [Apache Cassandra](#), [Apache Accumulo](#), [Amazon DynamoDB](#) and SQL databases such as [hsqldb](#), [MySQL](#) and more.

URI format

```
gora:instanceName[?options]
```

Hbase examples with mandatory options :

XML

```
<to uri="gora:foobar?keyClass=java.lang.Long&valueClass=org.apache.camel.component.gora.generated.
Pageview&dataStoreClass=org.apache.gora.hbase.store.HBaseStore" />
```

Java DSL

```
to("gora:foobar?keyClass=java.lang.Long&valueClass=org.apache.camel.component.gora.generated.
Pageview&dataStoreClass=org.apache.gora.hbase.store.HBaseStore")
```

Configuratiion

Using camel-gora needs some configuration. This mainly involve to configure the *AvroStore* through the *gora.properties* file and to define the relevant mappings as part of the *gora-core* module.

Extensive information for this configuration can be found in the [apache gora documentation](#) and the [gora-conf](#) page.

Supported Gora Operations

Supported operations include : *put*, *get*, *delete*, *getSchemaName*, *deleteSchema*, *createSchema*, *query*, *deleteByQuery*, *schemaExists*.

Some of the operations require arguments while some others no. The arguments to operations could be either the *body* of the *in* message or defined in a header property. Below there is a list with some additional info for each operation.

Property	Description
put	<i>Inserts the persistent object with the given key.</i>
get	<i>Returns the object corresponding to the given key fetching all the fields.</i>
delete	<i>Deletes the object with the given key.</i>
getSchemaName	<i>Returns the schema name given to this DataStore.</i>

deleteSchema	<i>Deletes the underlying schema or table (or similar) in the datastore that holds the objects.</i>
createSchema	<i>Creates the optional schema or table (or similar) in the datastore to hold the objects.</i>
query	<i>Executes the given query and returns the results.</i>
deleteByQuery	<i>Deletes all the objects matching the query.</i>
schemaExists	<i>Returns whether the schema that holds the data exists in the datastore.</i>

Options

Gora Headers

Property	Description
GoraOperation	<i>Used in order to define the operation to execute.</i>
GoraKey	<i>Used in order to define the datum key for the operations need it.</i>

Gora Configuration attributes

Property	Type	Description
keyClass	<i>String</i>	<i>Key type. *</i>
valueClass	<i>String</i>	<i>Value type. *</i>
dataStoreClass	<i>String</i>	<i>DataStore type*</i>
hadoopConfiguration	<i>Configuration</i>	<i>Hadoop Configuration</i>
concurrentConsumers	<i>int</i>	<i>Concurrent Consumers (used only by consumers).</i>
flushOnEveryOperation	<i>boolean</i>	<i>Flush on every operation (used only by producers).</i>

NOTE: the gora configuration properties marked with asterisk are mandatory

Gora Query attributes

Property	Type	Description
startTime	<i>long</i>	<i>Start Time attribute.</i>
endTime	<i>long</i>	<i>End Time attribute.</i>
timeRangeFrom	<i>long</i>	<i>Time Range From attribute.</i>
timeRangeTo	<i>long</i>	<i>Time Range To attribute.</i>
limit	<i>long</i>	<i>Gora Query Limit attribute.</i>
timestamp	<i>long</i>	<i>Timestamp attribute.</i>
startKey	<i>Object</i>	<i>Start Key attribute.</i>
endKey	<i>Object</i>	<i>End Key attribute.</i>
keyRangeFrom	<i>Object</i>	<i>Key Range From attribute.</i>
keyRangeTo	<i>Object</i>	<i>Key Range To attribute.</i>
fields	<i>String</i>	<i>Fields attribute.</i>

Usage examples

Create Schema (*XML DSL*):

```
<setHeader headerName="GoraOperation">
  <constant>CreateSchema</constant>
</setHeader>
<to uri="gora:foobar?keyClass=java.lang.Long&valueClass=org.apache.camel.component.gora.generated.
Pageview&dataStoreClass=org.apache.gora.hbase.store.HBaseStore" />
```

SchemaExists (XML DSL):

```
<setHeader headerName="GoraOperation">
  <constant>SchemaExists</constant>
</setHeader>
<to uri="gora:foobar?keyClass=java.lang.Long&valueClass=org.apache.camel.component.gora.generated.
Pageview&dataStoreClass=org.apache.gora.hbase.store.HBaseStore" />
```

Put (XML DSL):

```
<setHeader headerName="GoraOperation">
  <constant>put</constant>
</setHeader>
<setHeader headerName="GoraKey">
  <constant>22222</constant>
</setHeader>
<to uri="gora:foo?keyClass=java.lang.Long&valueClass=org.apache.camel.component.gora.generated.Pageview&
dataStoreClass=org.apache.gora.hbase.store.HBaseStore" />
```

Get (XML DSL):

```
<setHeader headerName="GoraOperation">
  <constant>GET</constant>
</setHeader>
<setHeader headerName="GoraKey">
  <constant>10101</constant>
</setHeader>
<to uri="gora:bar?keyClass=java.lang.Long&valueClass=org.apache.camel.component.gora.generated.Pageview&
dataStoreClass=org.apache.gora.hbase.store.HBaseStore" />
```

Delete (XML DSL):

```
<setHeader headerName="GoraOperation">
  <constant>DELETE</constant>
</setHeader>
<setHeader headerName="GoraKey">
  <constant>22222</constant>
</setHeader>
<to uri="gora:bar?keyClass=java.lang.Long&valueClass=org.apache.camel.component.gora.generated.Pageview&
dataStoreClass=org.apache.gora.hbase.store.HBaseStore" />
```

Query (XML DSL):

```
<to uri="gora:foobar?keyClass=java.lang.Long&valueClass=org.apache.camel.component.gora.generated.
Pageview&dataStoreClass=org.apache.gora.hbase.store.HBaseStore" />
```

The full usage examples in the form of integration tests can be found at [camel-gora-examples](#) repository.

More resources

For more please information and in depth configuration refer to the [Apache Gora Documentation](#) and the [Apache Gora Tutorial](#).