Couchbase

Available as of Camel Extra 2.13

The camel-couchbase component supports the interaction to the NoSQL document database Couchbase via the couchbase-client library. Couchbase is an high performance Document Store very easy to scale out, which supports topology changes with no downtime

Dependency

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

URI format

```
couchbase:http[s]://hostname/bucket?[options]
```

URI options

Name	Default Value	Type	Context	Description
additionalHosts	""	String	Shared	Comma separated list of additional hostnames
designDocumentName	"beer"	String	Consumer	The Document Design name, defaults to the beer example distributed with Couchbase
viewName	"brewery_beers"	String	Consumer	The Document view name, defaults, to the beer example distributed with Couchbase
limit	-1	int	Consumer	Limit the number of results that should be returned, default is unlimited
descending	false	boolean	Consumer	Revert the sorting order of the result set.
skip	0	int	Consumer	Number of results to skip
rangeStartKey	""	String	Consumer	Start Key to return records in the given key range.
rangeEndKey	""	String	Consumer	End Key to return records in the given key range.
consumerProcessedStrategy	none	String	Consumer	The strategy applied with consumed documents none = consumed documents are untouched, you should beware of duplicates (f.e. use an idempotent receiver) delete = consumed documents are deleted
operation	PUT	String	Producer	The operation to perform, currently supports the following values: PUT, DELETE, GET
autoStartIdForInserts	false	boolean	Producer	If set to true, document id will be automatically generated
startingIdForInsertsFrom	0	int	Producer	Starting value for the document id (if autoStartIdForInserts == true)
producerRetryAttempts	2	int	Producer	Number of retries
producerRetryPause	5000	int	Producer	Pause between retries (in ms)
persistTo	0	int	Producer	0: Don't wait for persistence on any nodes. 1: Persist to Master. 2: Persist to at least two nodes including Master. 3: Persist to at least three nodes including Master. 4: Persist to at least four nodes including Master.
replicateTo	0	int	Producer	0: Don't replicate. 1: Replicate to at least one node. 2: Replicate to at least two nodes. 3: Replicate to at least three nodes.

username	""	String	Shared	Username
password	""	String	Shared	Password

More URI options

Following URI options control how the CouchbaseConnectionFactoryBuilder instantiates the connection. Every option defaults to couchbase-client defaults.

Name	Default Value	Type	Context	Description
opTimeOut	2500	int	Shared	Time in milliseconds for an operation to time out
timeoutExceptionThreshold	998	int	Shared	Number of operations to time out before the node is deemed down
readBufferSize	16384	int	Shared	Read buffer size
maxReconnectDelay	30000	int	Shared	Maximum number of milliseconds to wait between reconnect attempts.
shouldOptimize	false	boolean	Shared	Optimize behavior for the network
opQueueMaxBlockTime	10000	int	Shared	The maximum time to block waiting for op queue operations to complete, in milliseconds.
obsPollInterval	400	int	Shared	Wait for the specified interval before the observe operation polls the nodes.
obsTimeout	-1	int	Shared	Observe operation timeout

Message Headers

Name	Default Value	Type	Context	Description
CCB_KEY	null	String	Shared	Key of the consumed row
CCB_ID	null	String	Consumer	Id of the consumed row
CCB_DDN	null	String	Consumer	Document Design name
CCB_VN	null	String	Consumer	View Name
CCB_TTL	0	int	Producer	TTL of the document

Example

Consume 10 documents from "beer-sample" bucket using Design Document "beer" and View "brewery_beers":

```
from("couchbase:http://localhost/beer-sample?designDocumentName=beer&viewName=brewery_beers&limit=10")
        .to("mock:result");
```

Add a document with ID "12346" to "default" bucket:

```
from("direct:start")
        .setHeader(CouchbaseConstants.HEADER_ID, constant("123456"))
        .to("couchbase:http://localhost/default");
```

Add documents to default bucket automatically generating ids, starting from 1000:

```
from("direct:start")
        .to("couchbase:http://localhost/default?autoStartIdForInserts=true&startingIdForInsertsFrom=1000")
```

Delete document with ID "120770" from "default" bucket:

```
from("direct:start")
        .setHeader(CouchbaseConstants.HEADER_ID, constant("120770"))
        .to("couchbase:http://localhost/default?operation='DELETE'")
```