Couchbase

Couchbase Java Client Library 1.1

Cheat Sheet

Connection Management

Connecting to Couchbase Server

private static **CouchbaseClient client**;
client.new CouchbaseClient([**url**] [, **urls**] [, **username**] [, **password**])

Parameters:

String url	URL for Couchbase Server Instance, or node
String urls	Linked list containing one or more URLs as strings
String username	Username for Couchbase bucket
String password	Password for Couchbase bucket

Disconnecting from Couchbase Server

client.shutdown();
client.shutdown(long **TimeValue**, **TimeUnit**);

Parameters:

long TimeValue	Wait value until any outstanding queued work is completed
enum TimeUnit	TimeUnit.SECONDS, TImeUnit.MINUTES

Store Operations

Add Operations

The add method adds a value to the database with the specified key, but will fail if the key already exists in the database.

client.add(**key**, **expiry**, **value**)
client.add(**key**, **expiry**, **value**, **transcoder**)

Replace Operations

The replace method will replace an existing key with a new value but will fail if the key does not exist in the database.

client.replace(**key**, **expiry**, **value**)
client.replace(**key**, **expiry**, **value**, **transcoder**)

Set Operations (including observe)

The set method stores a value to the database with a specified key.

client.set(**key**, **expiry**, **value**)
client.set(**key**, **expiry**, **value**, **transcoder**)
client.set(**key**, **expiry**, **value**, **persistto**)
client.set(**key**, **expiry**, **value**, **persistto, replicateto**)

Parameters:

String key	Key used to reference the value.
int expiry	Expiry time for the key in seconds. Values larger than 30*24*60*60 seconds (30 days) are interpreted as absolute times (from the epoch)
Object value	Value to be stored
enum persistto	Ability to specify persistence requirements - MASTER/ONE, TWO, THREE or FOUR nodes
enum replicateto	Ability to specify replication requirements – ZERO, ONE, TWO or THREE nodes
Transcoder<T> transcoder	Transcoder class to be used to serialize values

Retrieve Operations

There are several different flavors of retrieve operations in the Couchbase Server 2.0 Java SDK

client.asyncGetAndTouch(**key**, **expiry**)
client.asyncGetAndTouch(**key**, **expiry**, **transcoder**)
client.getAndTouch(**key**, **expiry**)
client.getAndTouch(**key**, **expiry**, **transcoder**)
client.asyncGet(**key**)
client.asyncGetBulk(**keycollection**)
client.asyncGetBulk(**keyn**)
client.asyncGetBulk(**transcoder**, **keyn**)
client.asyncGetBulk(**keycollection**, **transcoder**)
client.asyncGet(**key**, **transcoder**)
client.get(**key**)
client.getBulk(**keycollection**)
client.getBulk(**keyn**)
client.getBulk(**transcoder**, **keyn**)
client.getBulk(**keycollection**, **transcoder**)
client.get(**key**, **transcoder**)
client.asyncGetLock(**key [,getl-expiry]**)
client.asyncGetLock(**key [,getl-expiry]**, **transcoder**)
client.getAndLock(**key**, **getl-expiry**)
client.getAndLock(**key**, **getl-expiry**, **transcoder**)
client.asyncGets(**key**)
client.asyncGets(**key**, **transcoder**)
client.gets(**key**)
client.gets(**key**, **transcoder**)
client.unlock(**key**, **casunique**)

Parameters:

String key	Key used to reference the value. The key cannot contain control characters or whitespace
int expiry	Expiry time for the key in seconds. Values larger than 30*24*60*60 seconds (30 days) are interpreted as absolute times (from the epoch)
Collection<String> keycollection	One or more keys used to reference a value
String… keyn	One or more keys used to reference a value
long casunique	Unique value used to identify a key/value combination
int getl-expiry	Expiry time in seconds for lock : Default 15, Maximum 30
Transcoder<T> transcoder	Transcoder class to be used to serialize values

Statistics Operations

Obtain stats from all servers defined in a CouchbaseClient object
client.getStats()
client.getStats(**statname**)

Parameters:

String statname	Group name of a statistic for selecting individual statistic value

Did you know?

Couchbase Server 2.0 Beta is now available. You can get more information about the awesome features at:
http://www.couchbase.com/couchbase-server/beta

Couchbase

Couchbase Java Client Library 1.1

Cheat Sheet

Update Operations

The update methods support different methods of updating and changing existing information within Couchbase Server.

client.append(**casunique**, **key**, **value**)
client.append(**casunique**, **key**, **value**, **transcoder**)
client.asyncCAS(**key**, **casunique**, **value**)
client.asyncCAS(**key**, **casunique**, **expiry**, **value**, **transcoder**)
client.asyncCAS(**key**, **casunique**, **value**, **transcoder**)
client.cas(**key**, **casunique**, **value**)
client.cas(**key**, **casunique**, **expiry**, **value**, **transcoder**)
client.cas(**key**, **casunique**, **value**, **transcoder**)
client.asyncDecr(**key**, **offset**)
client.decr(**key**, **offset**)
client.decr(**key**, **offset**, **default**)
client.decr(**key**, **offset**, **default**, **expiry**)
client.delete(**key**)
client.asyncIncr(**key**, **offset**)
client.incr(**key**, **offset**)
client.incr(**key**, **offset**, **default**)
client.incr(**key**, **offset**, **default**, **expiry**)
client.prepend(**casunique**, **key**, **value**)
client.prepend(**casunique**, **key**, **value**, **transcoder**)
client.touch(**key**, **expiry**)

Parameters:

long casunique	Unique value used to identify a key/value combination
String key	Key used to reference the value. The key cannot contain control characters or whitespace
int offset	Integer offset value to increment / decrement (default is 1)
int default	Default value to increment/decrement if key does not exist
int expiry	Expiry time for the key in seconds. Values larger than 30*24*60*60 seconds (30 days) are interpreted as absolute times (from the epoch)
Object value	Value to be stored
Transcoder<T> transcoder	Transcoder class to be used to serialize values

Useful Links

Couchbase Website : http://www.couchbase.com

Couchbase Blog : http://blog.couchbase.com

Developer SDKs : http://www.couchbase.com/develop

Download: http://www.couchbase.com/download

Querying Operations

With Couchbase Server 2.0, you can add the power of views and querying those views to your applications.

Create a view object to be used when querying a view :
client.getView(**ddocname**, **viewname**)

Parameters:

String ddocname	Design document name
String viewname	View name within a design document

Then create a new query object to be used when querying the view
Query.new()
Query query = new Query();

Once, the view and query objects are available, the results of the server view can be accessed using
client.query(**view**, **query**)

Parameters:

View view	View object associated with a server view
Query query	Query object associated with a server view

Before accessing the view, a list of options can be set with the query object

setKey(**java.lang.String key**) to set a particular key in the B+ tree

setRangeStart(**java.lang.String startKey**) to set the starting key

setRangeEnd(**java.lang.String endKey**) to set the ending key

setRange(**java.lang.String startKey, java.lang.String endKey**) to set the starting and ending key, both together

setDescending(**boolean descending**) to set the descending flag to true or false

setIncludeDocs(**boolean include**) to include the original JSON document with the query

setReduce(**boolean reduce)** where the reduce function is executed based on the flag

setStale(**Stale reduce)** set to OK will not refresh the view even if it is stale, set to UPDATE_AFTER will update the view after the stale result is returned, FALSE will update the view and return the latest results.

The format of the returned information of the query method is : ViewResponse or any of the other inherited objects such as ViewResponseWithDocs, ViewResponseNoDocs, ViewResponseReduced

The ViewResponse method provides an iterator() method for iterating through the rows as a ViewRow interface. The ViewResponse method also provides a getMap() method where the result is available as a map.