

#01

test> db.test.help()

#### Collection Class:

aggregate	Calculates aggregate values for the data in a
collection or a view.	
bulkWrite	Performs multiple write operations with controls for
order of execution.	
count	Returns the count of documents that would match a
find() query for the collection or view.	
countDocuments	Returns the count of documents that match the query
for a collection or view.	
deleteMany	Removes all documents that match the filter from a
collection.	
deleteOne	Removes a single document from a collection.
distinct	Finds the distinct values for a specified field across
a single collection or view and returns the results in an array.	
estimatedDocumentCount	Returns the count of all documents in a collection or
view.	
find	Selects documents in a collection or view.
findAndModify	Modifies and returns a single document.
findOne	Selects documents in a collection or view.
renameCollection	Renames a collection.
findOneAndDelete	Deletes a single document based on the filter and sort
criteria, returning the deleted document.	
findOneAndReplace	Modifies and replaces a single document based on the
filter and sort criteria.	
findOneAndUpdate	Updates a single document based on the filter and sort
criteria.	
insert	Inserts a document or documents into a collection.
insertMany	Inserts multiple documents into a collection.
insertOne	Inserts a document into a collection.
isCapped	Checks if a collection is capped
remove	Removes documents from a collection.
replaceOne	Replaces a single document within the collection based
on the filter.	
update	Modifies an existing document or documents in a
collection.	
updateMany	Updates all documents that match the specified filter
for a collection.	
updateOne	Updates a single document within the collection based
on the filter.	
compactStructuredEncryptionData	Compacts structured encryption data
convertToCapped	calls {convertToCapped:'coll', size:maxBytes} command
createIndexes	Creates one or more indexes on a collection
createIndex	Creates one index on a collection
ensureIndex	Creates one index on a collection
getIndexes	Returns an array that holds a list of documents that
identify and describe the existing indexes on the collection.	
getIndexSpecs	Alias for getIndexes. Returns an array that holds a
list of documents that identify and describe the existing indexes on the collection.	
getIndices	Alias for getIndexes. Returns an array that holds a
list of documents that identify and describe the existing indexes on the collection.	
getIndexKeys	Return an array of key patterns for indexes defined on
collection	
dropIndexes	Drops the specified index or indexes (except the index
on the _id field) from a collection.	
dropIndex	Drops or removes the specified index from a
collection.	
totalIndexSize	Reports the total size used by the indexes on a
collection.	
reIndex	Rebuilds all existing indexes on a collection.

getDB	Get current database.
getMongo	Returns the Mongo object.
dataSize	This method provides a wrapper around the size output
of the collStats (i.e. db.collection.stats()) command.	
storageSize	The total amount of storage allocated to this
collection for document storage.	
totalSize	The total size in bytes of the data in the collection
plus the size of every index on the collection.	
drop	Removes a collection or view from the database.
exists	Returns collection infos if the collection exists or
null otherwise.	
getFullName	Returns the name of the collection prefixed with the
database name.	
getName	Returns the name of the collection.
runCommand	Runs a db command with the given name where the first
param is the collection name.	
explain	Returns information on the query plan.
stats	Returns statistics about the collection.
latencyStats	returns the \$latencyStats aggregation for the
collection. Takes an options document with an optional boolean 'histograms' field.	
initializeOrderedBulkOp	Initializes an ordered bulk command. Returns an
instance of Bulk	
initializeUnorderedBulkOp	Initializes an unordered bulk command. Returns an
instance of Bulk	
getPlanCache	Returns an interface to access the query plan cache
for a collection. The interface provides methods to view and clear the query plan cache.	
mapReduce	Calls the mapReduce command
validate	Calls the validate command. Default full value is
false	
getShardVersion	Calls the getShardVersion command
getShardDistribution	Prints the data distribution statistics for a sharded
collection.	
getShardLocation	Returns a document containing the shards where this
collection is located as well as whether the collection itself is sharded.	
watch	Opens a change stream cursor on the collection
hideIndex	Hides an existing index from the query planner.
unhideIndex	Unhides an existing index from the query planner.
analyzeShardKey	Returns metrics for evaluating a shard key. That is,
'key' can be a candidate shard key for an unsharded or sharded collection, or the current shard key	
for a sharded collection.	
configureQueryAnalyzer	Starts or stops collecting metrics about reads and
writes against an unsharded or sharded collection.	
checkMetadataConsistency	Returns a cursor with information about metadata
inconsistencies	
getSearchIndexes	Returns an array that holds a list of documents that
identify and describe the existing search indexes on the collection.	
createSearchIndex	Creates one search indexes on a collection
createSearchIndexes	Creates one or more search indexes on a collection
dropSearchIndex	Drops or removes the specified search index from a
collection.	
updateSearchIndex	Updates the sepecified search index.