

```
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 query.
find() query for the collection or view.	Returns the count of documents that match the query for a collection or view.
countDocuments	Removes all documents that match the filter from a collection.
for a collection or view.	Removes a single document from a collection.
deleteMany	Finds the distinct values for a specified field across a single collection or view and returns the results in an array.
collection.	Returns the count of all documents in a collection or view.
deleteOne	Selects documents in a collection or view.
distinct	Modifies and returns a single document.
a single collection or view and returns the results in an array.	Selects documents in a collection or view.
estimatedDocumentCount	Renames a collection.
view.	Deletes a single document based on the filter and sort criteria, returning the deleted document.
find	Modifies and replaces a single document based on the filter and sort criteria.
findAndModify	Updates a single document based on the filter and sort criteria.
findOne	Inserts a document or documents into a collection.
renameCollection	Inserts multiple documents into a collection.
findOneAndDelete	Inserts a document into a collection.
criteria, returning the deleted document.	Checks if a collection is capped.
findOneAndReplace	Removes documents from a collection.
filter and sort criteria.	Replaces a single document within the collection based on the filter.
findOneAndUpdate	Modifies an existing document or documents in a collection.
criteria.	Updates all documents that match the specified filter for a collection.
insert	Updates a single document within the collection based on the filter.
insertMany	Compacts structured encryption data.
insertOne	calls {convertToCapped:'coll', size:maxBytes} command.
isCapped	Creates one or more indexes on a collection.
remove	Creates one index on a collection.
replaceOne	Creates one index on a collection.
on the filter.	Returns an array that holds a list of documents that identify and describe the existing indexes on the collection.
update	Alias for getIndexes. Returns an array that holds a list of documents that identify and describe the existing indexes on the collection.
collection.	Alias for getIndexes. Returns an array that holds a list of documents that identify and describe the existing indexes on the collection.
updateMany	getIndexSpecs
for a collection.	list of documents that identify and describe the existing indexes on the collection.
updateOne	getIndices
on the filter.	list of documents that identify and describe the existing indexes on the collection.
compactStructuredEncryptionData	
convertToCapped	
createIndexes	
createIndex	
ensureIndex	
getIndexes	
identify and describe the existing indexes on the collection.	
getIndexSpecs	
list of documents that identify and describe the existing indexes on the collection.	
getIndices	
list of documents that identify and describe the existing indexes on the collection.	

```
test> db.help()
```

Database Class:

getMongo	Returns the current database connection
getName	Returns the name of the DB
getCollectionNames	Returns an array containing the names of all collections in the current database.
getCollectionInfos	Returns an array of documents with collection

information, i.e. collection name and options, for the current database.

runCommand	Runs an arbitrary command on the database.
adminCommand	Runs an arbitrary command against the admin database.
aggregate	Runs a specified admin/diagnostic pipeline which does not require an underlying collection.

not require an underlying collection.

getSiblingDB	Returns another database without modifying the db variable in the shell environment.
--------------	--

variable in the shell environment.

getCollection	Returns a collection or a view object that is functionally equivalent to using the db.<collectionName>.
---------------	---

getCollection

dropDatabase	Removes the current database, deleting the associated data files.
--------------	---

createUser

method is run. db.createUser() returns a duplicate user error if the user already exists on the database.	Creates a new user for the database on which the method is run. db.createUser() returns a duplicate user error if the user already exists on the database.
---	--

updateUser

you run the method. An update to a field completely replaces the previous field's values. This includes updates to the user's roles array.	Updates the user's profile on the database on which you run the method. An update to a field completely replaces the previous field's values. This includes updates to the user's roles array.
--	--

changeUserPassword

database where the user is defined, i.e. the database you created the user.	Updates a user's password. Run the method in the database where the user is defined, i.e. the database you created the user.
---	--

logout

has no effect if the current session is not authenticated.	Ends the current authentication session. This function has no effect if the current session is not authenticated.
--	---

dropUser

dropAllUsers	Removes the user from the current database.
--------------	---

auth

within the shell.	Removes all users from the current database.
-------------------	--

grantRolesToUser

revokeRolesFromUser	Allows a user to authenticate to the database from database.
---------------------	--

database.

getUser

this method on the user's database. The user must exist on the database on which the method runs.	Returns user information for a specified user. Run this method on the user's database. The user must exist on the database on which the method runs.
---	--

getUsers

createCollection	Returns information for all the users in the database.
------------------	--

createEncryptedCollection

fields each with unique and auto-created data encryption keys (DEKs). This is a utility function that internally utilises ClientEncryption.createEncryptedCollection.	Create new collection
---	-----------------------

createView

createRole	Creates a new collection with a list of encrypted fields each with unique and auto-created data encryption keys (DEKs). This is a utility function that internally utilises ClientEncryption.createEncryptedCollection.
------------	---

updateRole

you run the method. An update to a field completely replaces the previous field's values.	Updates the role's profile on the database on which you run the method. An update to a field completely replaces the previous field's values.
---	---

dropRole

dropAllRoles	Removes the role from the current database.
--------------	---

grantRolesToRole

revokeRolesFromRole	Removes all roles from the current database.
---------------------	--

database.

createView

createRole	Grants additional roles to a role.
------------	------------------------------------

updateRole

you run the method. An update to a field completely replaces the previous field's values.	Removes a one or more roles from a role on the current database.
---	--

dropRole

dropAllRoles	Creates a new role.
--------------	---------------------

grantRolesToRole

revokeRolesFromRole	Updates the role's profile on the database on which you run the method. An update to a field completely replaces the previous field's values.
---------------------	---

createView

createRole	Removes the role from the current database.
------------	---

updateRole

you run the method. An update to a field completely replaces the previous field's values.	Grants additional roles to a role.
---	------------------------------------

dropRole

dropAllRoles	Removes all roles from the current database.
--------------	--

grantRolesToRole

revokeRolesFromRole	Creates a new role.
---------------------	---------------------

database.

grantPrivilegesToRole

revokePrivilegesFromRole	Updates the role's profile on the database on which you run the method. An update to a field completely replaces the previous field's values.
--------------------------	---

createView

createRole	Removes the role from the current database.
------------	---

updateRole

you run the method. An update to a field completely replaces the previous field's values.	Grants additional privileges to a role.
---	---

dropRole

dropAllRoles	Removes all roles from the current database.
--------------	--

grantRolesToRole

revokeRolesFromRole	Creates a new role.
---------------------	---------------------

database.

grantPrivilegesToRole

revokePrivilegesFromRole	Updates the role's profile on the database on which you run the method. An update to a field completely replaces the previous field's values.
--------------------------	---

createView

createRole	Removes the role from the current database.
------------	---

updateRole

you run the method. An update to a field completely replaces the previous field's values.	Grants additional privileges to a role.
---	---

dropRole

dropAllRoles	Removes all roles from the current database.
--------------	--

grantRolesToRole

revokeRolesFromRole	Creates a new role.
---------------------	---------------------

database.

getRole

this method on the role's database. The role must exist on the database on which the method runs.	Returns role information for a specified role. Run this method on the role's database. The role must exist on the database on which the method runs.
---	--

getRoles

currentOp	Returns information for all the roles in the database.
-----------	--

a document that contains information on in-progress operations for the database instance. For further information, see \$currentOp.

killOp

specified by the operation ID. To find operations and their corresponding IDs, see \$currentOp or db.currentOp().	Calls the killOp command. Terminates an operation as specified by the operation ID. To find operations and their corresponding IDs, see \$currentOp or db.currentOp().
---	--

shutdownServer

mongod or mongos process cleanly and safely. You must issue the db.shutdownServer() operation against the admin database.	Calls the shutdown command. Shuts down the current mongod or mongos process cleanly and safely. You must issue the db.shutdownServer() operation against the admin database.
---	--

fsyncLock

all pending write operations to disk and locks the entire mongod instance to prevent additional writes until the user releases the lock with a corresponding db.fsyncUnlock() command.	Calls the fsync command. Forces the mongod to flush all pending write operations to disk and locks the entire mongod instance to prevent additional writes until the user releases the lock with a corresponding db.fsyncUnlock() command.
--	--

fsyncUnlock

by db.fsyncLock() on a mongod instance by 1.	Calls the fsyncUnlock command. Reduces the lock taken by db.fsyncLock() on a mongod instance by 1.
--	--

version

	returns the db version. uses the buildinfo command
--	--

```
serverBits           returns the db serverBits. uses the buildInfo command
isMaster             Calls the isMaster command
hello                Calls the hello command
serverBuildInfo      returns the db serverBuildInfo. uses the buildInfo
command
serverStatus         returns the server stats. uses the serverStatus
command
stats               returns the db stats. uses the dbStats command
hostInfo             Calls the hostInfo command
serverCmdLineOpts   returns the db serverCmdLineOpts. uses the
getCmdLineOpts      command
rotateCertificates  Calls the rotateCertificates command
printCollectionStats Prints the collection.stats for each collection in the
db.
getProfilingStatus  returns the db getProfilingStatus. uses the profile
command
setProfilingLevel   returns the db setProfilingLevel. uses the profile
command
setLogLevel          returns the db setLogLevel. uses the setParameter
command
getLogComponents    returns the db getLogComponents. uses the getParameter
command
cloneDatabase        deprecated, non-functional
cloneCollection      deprecated, non-functional
copyDatabase         deprecated, non-functional
commandHelp          returns the db commandHelp. uses the passed in command
with help: true
listCommands          Calls the listCommands command
getLastErrorObj      Calls the getLastError command
getLastError          Calls the getLastError command
printShardingStatus  Calls sh.status(verbose)
printSecondaryReplicationInfo Prints secondary replicaset information
getReplicationInfo   Returns replication information
printReplicationInfo Formats sh.getReplicationInfo
printSlaveReplicationInfo DEPRECATED. Use db.printSecondaryReplicationInfo
setSecondaryOk        This method is deprecated. Use
db.getMongo().setReadPref() instead
watch                Opens a change stream cursor on the database
sql                  (Experimental) Runs a SQL query against Atlas Data
Lake. Note: this is an experimental feature that may be subject to change in future releases.
checkMetadataConsistency Returns a cursor with information about metadata
inconsistencies
```