

Collection Management 📤

Now after we have learned how to manipulate a database, it's time to manipulate the collections of ur database:

P.S. The methods listed on this table of contents page refer to the mongo shell methods, and not to the MongoDB Node.js driver (or any other driver) methods.

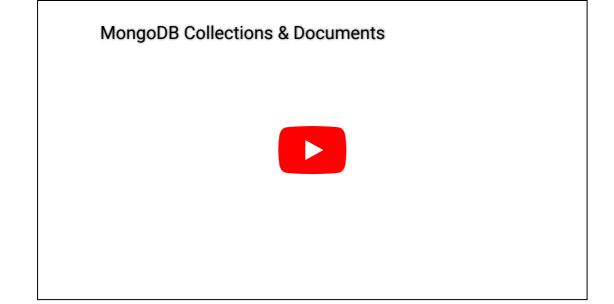
- db.createCollection(): Because MongoDB creates a collection implicitly when the collection is first referenced in a command, this method is used primarily for creating new collections that use specific options. For example, you use db.createCollection() to create a capped collection, or to create a new collection that uses document validation.
- db.collection.count(): Returns the documents count that would match a find() query for the collection or view. The db.collection.count() method does not perform the find() operation but instead counts and returns the number of results that match a query.

Collection Management



- db.collection.stats(): Reports a collection's state.
- db.collection.storageSize(): Reports the total storage size used by the collection in bytes.
- db.collection.validate(<option>): Validates a collection. The method scans a collection data and indexes for correctness and returns the result.
 - o full: Optional. A flag that determines whether the command performs a slower but more thorough check, or a faster but less thorough check.

```
db.collection.validate( {
   full: <boolean> // Optional
} )
db.collection.validate( <boolean> ) // full option
```



Collection Management 📤

cb.collection.explain(verbosity)

Returns information on the query plan for the following methods:

- aggregate()
- count()
- find()

- remove()
- update()
- distinct()
- findAndModify()

Verbosity: Optional. Specifies the verbosity mode for the explain output. The mode affects the behavior of explain() and determines the amount of information to return. The possible modes are:

- "queryPlanner" (Default)
- "executionStats"
- "allPlansExecution"

< Previous

next >

Ш

P

*

0

©