



Mongo DB- Insert, Delete, Drop

By William Lawrence



Insert

- Inserts are the basic method for adding data to MongoDB
- There are multiple insert methods, when inserting a single document use the **insertOne** method.

```
try {  
    db.products.insertOne( { item: "card", qty: 15 } );  
} catch (e) {  
    print (e);  
};
```

- The code above inserts the specified document into the database products
- insertOne also has a “_id” key, that when not specified Mongo will add one.



```
try {
    db.products.insertOne({_id: 10, item: "box", qty: 20});
} catch (e) {
    print (e);
}
```

Can not insert items with the same id



insertMany

- When needing to insert multiple documents into a collection, **insertMany** is used allowing an array of documents to be passed to the database.
- More efficient because it allows insertion in bulk
- Some Caveats
 - There is a data limit to how much can be inserted at once, no more than 48mb. Otherwise Mongo will split the batch insert into multiples.
 - There is second parameter “ordered” that determines the whether the documents are inserted in the order provided or not.
 - When true if a document produces an insertion error, no documents beyond that point will be inserted, while a unordered insert, inserts regardless.



Delete

- Two methods for deletion. `deleteOne` and `deleteMany`. Functions similar to `insert`. However, both of these methods take a filter document as their first parameter.
- The filter specifies a set of criteria to match against in removing documents.
- `deleteOne` deletes the first document to match the filter
 - `db.movies.deleteOne({ "_id" : 4 })`
- When, wanting to clear an entire collection use **drop**.
 - `db.movies.drop()`



Creating Collections

`db.createCollection()`

- Takes two parameters
 - Name
 - The name of the collection to create
 - Options
 - Optional. Configuration options for creating a capped collection, for preallocating space in a new collection, or for creating a view.



To create a time series collection that captures weather data for the past 24 hours, issue this command:

```
db.createCollection(
  "Weather24h",
  {
    timeseries: {
      timeField: "timestamp",
      metaField: "data",
      granularity: "hours"
    },
    expireAfterSeconds: 86400
  }
)
```



Creating DataBase

use DATABASE_NAME

The command will create a new database if it doesn't exist, otherwise it will return the existing database.