

## Introduction to the MongoDB deleteOne() method

The deleteOne() method allows you to delete a single document from a collection.

The deleteOne() method has the following syntax:

```
db.collection.deleteOne(filter, option)
```

Code language: CSS (css)

The deleteOne() method accepts two arguments:

- filter is an required argument. The filter is a document that specifies the condition for matching the document to remove. If you pass an empty document {} into the method, it'll delete the first document in the collection.
- option is an optional argument. The option is a document that specifies the deletion options.

The deleteOne() method returns a document containing the deleteCount field that stores the number of deleted documents.

To delete all documents that match a condition from a collection, you use the deleteMany() method.

## MongoDB deleteCount() method examples

We'll use the following products collection:

```
db.products.insertMany([
  { "_id" : 1, "name" : "xPhone", "price" : 799, "releaseDate": ISODate("2011-05-14"), "spec"
  : { "ram" : 4, "screen" : 6.5, "cpu" : 2.66 }, "color":["white","black"],"storage":[64,128,256]},
  { "_id" : 2, "name" : "xTablet", "price" : 899, "releaseDate": ISODate("2011-09-01") , "spec"
  : { "ram" : 16, "screen" : 9.5, "cpu" : 3.66
  }, "color":["white","black","purple"],"storage":[128,256,512]},
  { "_id" : 3, "name" : "SmartTablet", "price" : 899, "releaseDate": ISODate("2015-01-14"),
  "spec" : { "ram" : 12, "screen" : 9.7, "cpu" : 3.66 }, "color":["blue"],"storage":[16,64,128]},
  { "_id" : 4, "name" : "SmartPad", "price" : 699, "releaseDate": ISODate("2020-05-
  14"), "spec" : { "ram" : 8, "screen" : 9.7, "cpu" : 1.66
  }, "color":["white","orange","gold","gray"],"storage":[128,256,1024]},
  { "_id" : 5, "name" : "SmartPhone", "price" : 599, "releaseDate": ISODate("2022-09-14"),
  "spec" : { "ram" : 4, "screen" : 9.7, "cpu" : 1.66
  }, "color":["white","orange","gold","gray"],"storage":[128,256]}
])
```

Code language: JavaScript (javascript)

### 1) Using the deleteOne() method to delete a single document

The following example uses the deleteOne() method to delete a document with the \_id is 1 from the products collection:

```
db.products.deleteOne({ _id: 1 })
```

Code language: CSS (css)

The query returned the following document:

```
{ "acknowledged" : true, "deletedCount" : 1 }
```

Code language: JSON / JSON with Comments (json)

The deleteCount field returned 1 indicating the number of deleted documents.

### 2) Using the deleteOne() method to delete the first document from a collection

The following query uses the deleteOne() method to remove the first document returned from the products collection:

```
db.products.deleteOne({})
```

Code language: CSS (css)

It returned the following document:

```
{ "acknowledged" : true, "deletedCount" : 1 }
```

Code language: JSON / JSON with Comments (json)

In this example, we passed an empty document {} to the deleteOne() method. Therefore, it removed the first document from the products collection.

### Summary

- Use the MongoDB deleteOne() method to remove a single document that matches a condition.
- Pass an empty document into the deleteOne() method to delete the first document from the collection.

### Introduction to MongoDB deleteMany() method

The deleteMany() method allows you to remove all documents that match a condition from a collection.

The deleteMany() has the following syntax:

```
db.collection.deleteMany(filter, option)
```

Code language: CSS (css)

In this syntax:

- filter is a document that specifies deletion criteria. If the filter is an empty document {}, the deleteMany() method will delete all documents from the collection.
- option is a document that specifies the deletion option.

The deleteMany() returns a document containing the deleteCount field that stores the number of deleted documents.

To delete a single document from a collection, you can use the [deleteOne\(\)](#) method.

MongoDB deleteMany() method examples

We'll use the following products collection.

```
db.products.insertMany([
  { "_id" : 1, "name" : "xPhone", "price" : 799, "releaseDate": ISODate("2011-05-14"), "spec"
: { "ram" : 4, "screen" : 6.5, "cpu" : 2.66 }, "color":["white","black"],"storage":[64,128,256]},
  { "_id" : 2, "name" : "xTablet", "price" : 899, "releaseDate": ISODate("2011-09-01") , "spec"
: { "ram" : 16, "screen" : 9.5, "cpu" : 3.66
}, "color":["white","black","purple"],"storage":[128,256,512]},
  { "_id" : 3, "name" : "SmartTablet", "price" : 899, "releaseDate": ISODate("2015-01-14"),
"spec" : { "ram" : 12, "screen" : 9.7, "cpu" : 3.66 }, "color":["blue"],"storage":[16,64,128]},
  { "_id" : 4, "name" : "SmartPad", "price" : 699, "releaseDate": ISODate("2020-05-
14"),"spec" : { "ram" : 8, "screen" : 9.7, "cpu" : 1.66
}, "color":["white","orange","gold","gray"],"storage":[128,256,1024]},
  { "_id" : 5, "name" : "SmartPhone", "price" : 599, "releaseDate": ISODate("2022-09-14"),
"spec" : { "ram" : 4, "screen" : 9.7, "cpu" : 1.66
}, "color":["white","orange","gold","gray"],"storage":[128,256]}
])
```

Code language: JavaScript (javascript)

1) Using the deleteMany() method to delete multiple documents

The following example uses the deleteMany() method to delete all documents where the value in the price field is 899:

```
db.products.deleteMany({ price: 899 })
```

Code language: CSS (css)

In this example, the filter argument is `{ price: 899 }` that matches documents whose price is 899.

It returned the following document:

```
{ "acknowledged" : true, "deletedCount" : 2 }
```

Code language: JSON / JSON with Comments (json)

The deleteCount field indicates that the number of deleted documents is 2.

## 2) Using the deleteMany() method to delete all documents

The following query uses the deleteMany() method to delete all documents from the products collection:

```
db.products.deleteMany({})
```

Code language: CSS (css)

Output:

```
{ acknowledged: true, deletedCount: 3 }
```

Code language: CSS (css)

In this example, we passed an empty document `{}` into the deleteMany() method. Therefore, it deleted all documents from the products collection.

## Summary

- Use the deleteMany() method to remove all documents that match a condition from a collection.
- Pass an empty document `{}` into the deleteMany() method to remove all documents from the collection.