

Node.js – Delete Collection in MongoDB

Node.js – Delete Collection in MongoDB

In this [Node.js Tutorial](#), we shall learn to Delete Collection in [MongoDB](#) from Node.js Application, using `db.collection.remove()` method, with an example.

Following is a step by step guide with an example to delete a collection in MongoDB from Node.js Application.

1. Start MongoDB Service. Run the following command to start MongoDB Service

```
sudo service mongod start
```

```
sudo service mongod start
```

2. Get the base URL to MongoDB Service. A simple hack to know the base url of MongoDB Service is to Open a Terminal and run [Mongo Shell](#).

```
Terminal - Mongo Shell
```

```
arjun@nodejs:~$ mongo
```

```
MongoDB shell version v3.4.9
```

```
connecting to: mongodb://127.0.0.1:27017
```

```
MongoDB server version: 3.4.9
```

```
Server has startup warnings:
```

```
2017-10-29T18:15:36.110+0530 | STORAGE [initandlisten]
```

While the Mongo Shell starts up, it echoes back the base url of MongoDB.

```
mongodb://127.0.0.1:27017
```

```
mongodb://127.0.0.1:27017
```

3. Prepare the complete URL. Append the Database name you want to connect to (say **newdb**), to the base URL.

```
mongodb://127.0.0.1:27017/newdb
```

```
mongodb://127.0.0.1:27017/newdb
```

4. Create a MongoClient.

```
var MongoClient =  
require('mongodb').MongoClient
```

```
var MongoClient = require('mongodb').MongoClient;
```

5. Make connection from MongoClient to the MongoDB Server with the help of URL.

```
MongoClient.connect(u
```

```
MongoClient.connect(url, <callback_function>);
```

Once the MongoClient is done trying to make a connection, the **callback function** receives error and db object as arguments.

If the connection is successful, the db object points to the database, **newdb**.

6. Get reference to MongoDB Collection.

```
db.collection(<collectio
```

```
db.collection(<collection_name>, <callback_function>);
```

Once you get db object that points to the specified mongodb database, use it to get reference to required collection using the above statement.

7. **Delete MongoDB Collection.** Following is the syntax of remove() method used to delete collection in MongoDB from Node.js.

```
collection.remove({}, call
```

```
collection.remove({}, callback_function)
```

| | |
|--------------------------|---|
| collection | reference to the mongodb collection that we would like to delete |
| callback_function | This Node.js Callback Function is called after Node has tried deleting the specified collection, and ready with the result. The callback function receives error and result object as arguments. |

Example Node.js program

node-js-mongodb-delete-collection.js

```
// example : delete  
local collection in
```

```

// example : delete 'users' collection in newdb database
var url = "mongodb://localhost:27017/newdb";

// create a client to mongodb
var MongoClient = require('mongodb').MongoClient;

// make client connect to mongo service
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  // db pointing to newdb
  console.log("Switched to "+db.databaseName+" database");
  // get reference to collection
  db.collection("users", function(err, collection) {
    // handle the error if any
    if (err) throw err;
    // delete the mongodb collection
    collection.remove({}, function(err, result){
      // handle the error if any
      if (err) throw err;
      console.log("Collection is deleted! "+result);
      // close the connection to db when you are done with it
      db.close();
    });
  });
});

```

Output

```

~$ node node-js-
mongodb_delete

```

```

~$ node node-js-mongodb-delete-collection.js
Switched to newdb database
Collection is deleted! {"n":0,"ok":1}

```

Reference

[MongoDB Tutorial](#) – Learn MongoDB from basics with Examples.

Conclusion :

In this [Node.js MongoDB](#) tutorial : **Node.js – Delete Collection in MongoDB** we have learnt to delete a collection from MongoDB Database with Node.js Application using mongodb package. In our next tutorial – Node.js Insert Document to MongoDB Collection, we shall learn to insert one or more documents to a MongoDB collection.

Node.js

▸ [Node.js Tutorial](#)

Get Started With Node.js

▸ [Install Node.js Ubuntu Linux](#)

▸ [Install Node.js Windows](#)

▸ [Node.js - Basic Example](#)

▸ [Node.js - Command Line Arguments](#)

▸ [Node.js - Modules](#)

▸ [Node.js - Create a module](#)

▸ [Node.js - Add new functions to Module](#)

▸ [Node.js - Override functions of Module](#)

▸ [Node.js - Callback Function](#)

▸ [Node.js - forEach](#)

Express.js

▸ [Express.js Tutorial](#)

▸ [What is Express.js?](#)

▸ [Express.js Application Example](#)

▸ [Install Express.js](#)

▸ [Express.js Routes](#)

▸ [Express.js Middleware](#)

▸ [Express.js Router](#)

Node.js Buffers

▸ [Node.js Buffer - Create, Write, Read](#)

▸ [Node.js Buffer - Length](#)

▸ [Node.js - Convert JSON to Buffer](#)

▸ [Node.js - Array to Buffer](#)

Node.js HTTP

▸ [Node.js - Create HTTP Web Server](#)

▸ [Node.js - Redirect URL](#)

Node.js MySQL

▸ [Node.js MySQL](#)

▸ [Node.js MySQL - Connect to MySQL Database](#)

▸ [Node.js MySQL - SELECT FROM](#)

Node.js MySQL - SELECT WHERE

Node.js MySQL - ORDER BY

Node.js MySQL - INSERT INTO

Node.js MySQL - UPDATE

Node.js MySQL - DELETE

Node.js MySQL - Result Object

Node.js MongoDB

Node.js MongoDB

Node.js - Connect to MongoDB

Node.js - Create Database in MongoDB

Node.js - Drop Database in MongoDB

Node.js - Create Collection in MongoDB

Node.js - Delete Collection in MongoDB

Node.js - Insert Documents to MongoDB Collection

MongoError: failed to connect to server

Node.js Mongoose

Node.js Mongoose Tutorial

Node.js Mongoose - Installation

Node.js Mongoose - Connect to MongoDB

Node.js Mongoose - Define a Model

Node.js Mongoose - Insert Single Document to MongoDB

Node.js Mongoose - Insert Multiple Documents to MongoDB

Node.js URL

Node.js - Parse URL parameters

Node.js FS (File System)

Node FS

Node FS - Read a File

Node FS - Create a File

Node FS - Write to a File

Node FS - Append to a File

Node FS - Rename a File

Node FS - Delete a File

Node FS Extra - Copy a Folder

Node.js JSON

‡ [Node.js Parse JSON](#)

‡ [Node.js Write JSON Object to File](#)

Node.js Error Handling

‡ [Node.js Try Catch](#)

Node.js Examples

‡ [Node.js Examples](#)

‡ [Node.js - Handle Get Requests](#)

‡ [Node.js Example - Upload files to Node.js server](#)

Useful Resources

‡ [Node.js Interview Questions](#)

‡ [How to Learn Programming](#)