Node.js – Create Database in MongoDB

Node.js – Create Database in MongoDB

In this Node.js Tutorial, we shall learn to Create Database in MongoDB from Node.js Application with an example.

Example

Following is a step by step guide with an example to create a database 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. Install mongodb package using npm. Refer Node.js MongoDB Tutorial to install mongodb package.
3. 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
```

```
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 I STORAGE [initandlisten]
```

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

```
mongodb://127.0.0.1:2
7017
```

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

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

```
mongodb://127.0.0.1:2
7017/newdb
```

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

5. Create a MongoClient.

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

6. Make connection from MongoClient to the MongoDB Server with the help of URL. [Note : In MongoDB, creating Database is an implicit process.]

```
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 newly created database **newdb**.

Example Node.js program

node-js-mongodb-create-database.js

```
// newdb is the new database we create
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;
    console.log("Database created!");
    // print database name
    console.log("db object points to the database : "+ db.databaseName);
    // after completing all the operations with db, close it.
    db.close();
});
```

Output

```
arjun@tutorialkart:~/workspace/nodejs/mong
```

```
arjun@tutorialkart:~/workspace/nodejs/mongodb$ node node-js-mongodb-create-database.js
Database created!
db object points to the database : newdb
```

Reference

MongoDB Tutorial – Learn MongoDB from basics with Examples.

Conclusion :

In this Node.js MongoDB tutorial : **Node.js – Create Database in MongoDB**, we have learnt to create a database from Node.js Application using mongodb package. In our next tutorial – Node.js MongoDB Drop Database, we shall learn to delete the database.

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