

Node.js Mongoose – Connect to MongoDB

Node.js Mongoose – Connect to MongoDB

To connect to MongoDB from Node.js using Mongoose package, call `connect()` function, on the variable referencing to mongoose, with MongoDB Database URI passed as argument to the function.

```
var mongoose =  
require('mongoose');  
  
var mongoose = require('mongoose');  
mongoose.connect('mongodb://localhost:27017/database_name');
```

Get reference to Database

To get a reference to the database specified, use `connection()` function on Mongoose reference, as shown below :

```
var db =  
mongoose.connection;  
  
var db = mongoose.connection;
```

Is connection successful ?

To check if the connection is successful or not, you may use callback functions : `on()` and `once()`.

```
var mongoose =  
require('mongoose');  
  
var mongoose = require('mongoose');  
mongoose.connect('mongodb://localhost:27017/tutorialkart');  
  
var db = mongoose.connection;  
  
db.on('error', console.error.bind(console, 'connection error:'));  
  
db.once('open', function() {  
  console.log("Connection Successful!");  
});
```

```
$ node node-js-  
mongodb-connection.js
```

```
$ node node-js-mongodb-connection.js  
Connection Successful!
```

Connection is successful.

To simulate 'Connection not successful' scenario, let's change the port to some incorrect value.

```
var mongoose =  
require('mongoose');
```

```
var mongoose = require('mongoose');  
# incorrect port number  
mongoose.connect('mongodb://localhost:ab017/tutorialkart');  
  
var db = mongoose.connection;  
  
db.on('error', console.error.bind(console, 'connection error:'));  
  
db.once('open', function() {  
  console.log("Connection Successful!");  
});
```

If the connection is not successful, you may see the following error message displayed on console.

```
$ node node-js-  
mongodb-connection.js
```

```
$ node node-js-mongodb-connection.js  
(node:8986) UnhandledPromiseRejectionWarning: Unhandled promise rejection (rejection id: 2): Error: Slash in host identifier  
(node:8986) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections  
that are not handled will terminate the Node.js process with a non-zero exit code.
```

When the connection is made without any errors and is open, callback function provided to `db.open('open', callback)` is executed.

Conclusion:

In this tutorial – Node.js **Mongoose – Connect to MongoDB** we have learnt to connect to MongoDB using Mongoose, get the reference to Database, check if the connection is successful or not.

‡ [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](#)