

Node.js MySQL Connect to DATABASE

Learn to Connect to MySQL database in Node.js using `mysql.createConnection` method with an example Node.js program.

Node.js MySQL Connect to a database

Node.js provides a module to connect to MySQL database and perform MySQL operations.



Following is a step-by-step guide to connect to MySQL database with an example.

- Step 1 : Make sure that MySQL is configured properly. Collect the information about IP Address, User Name and Password.
- Step 2 : In your node.js program, include mysql module.

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

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

MySQL module has to be installed before it can be used. Otherwise you would get an error. To install mysql module, refer [how to install MySQL in Node.js](#).

- Step 3 : Create a connection variable with IP Address, User Name and Password collected in step 1.

```
var con =  
mysql.createConnecti
```

```
var con = mysql.createConnection({  
  host: "localhost", // ip address of server running mysql  
  user: "arjun", // user name to your mysql database  
  password: "password" // corresponding password  
});
```

- Step 4 : Make a call to connect function using connection variable, that we created in the previous step.

```
con.connect(function  
(err) {
```

```
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
});
```

Function provided as argument to connect is a callback function. After node.js has tried with connecting to MySQL Database, the callback function is called with the resulting information sent as argument to the callback function.

Example program to connect to MySQL database in Node.js

connectToMySQL.js - Connect to MySQL database in Node.js

```
// include mysql module
var mysql = require('mysql');

// create a connection variable with the details required
var con = mysql.createConnection({
  host: "localhost", // ip address of server running mysql
  user: "arjun", // user name to your mysql database
  password: "password" // corresponding password
});

// connect to the database.
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
});
```

```
$ node
connectToMySQL.js
Connected!
```

Conclusion :

In this [Node.js Tutorial](#) – [Node.js MySQL](#) – Connect to Database, we have learnt to Connect to MySQL database using Node.js MySQL Module ->**createConnection** method with the help of an example Node.js program.

‣ [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 - 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](#)