# Setting DB Connection From Node JS

**This guide includes steps to install different DB connectors for Node JS**

* **For My SQL Server – mysql npm package**
* **For Oracle Server - oracledb npm package**
* **For MS SQL Server – mssql npm package**
* **For Mongo DB - mongoose npm package**

This guide also contains sample code to connect each DB using respective npm package.

1. **My SQL Server – mssql**

Please execute below command to install My sql DB connector.

**npm install --save mysql**

Example program to connect DB using mysql connector:-

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

**var** mysqldbconfig = {

host : 'localhost',

user : 'root',

password : 'root',

database : 'employee'

};

**var** connecton = mysql.createConnection(mysqldbconfig);

connecton.connect(**function**(err) {

**if** (err) {

console.log('Error connecting to Db');

**return**;

}

console.log('Connection established');

});

**var** selectAll = connecton.query('select \* from employee ',

**function**(err, result) {

**if** (err) {

console.error(err);

} **else** {

console.log('Select All ids');

console.log(result);

/\*for (var i = 0; i < result.length; i++) {

console.log("Id : "+result[i].id);

console.log("Name : "+result[i].name);

};\*/

}

**return** result;

});

1. **Oracle Server – mssql**

Please execute below command to install Oracle DB connector.

**npm install --save https://github.com/bchr02/node-oracledb#v1.8.0**

Example program to connect DB using oracledb connector:-

**var** oracledb = require('oracledb');

**var** oracledbconfig = {

user : 'a\*\*\*\*\*',

password : 'a\*\*\*\*\*',

connectString : '10.78.145.235/CT'

};

oracledb.getConnection(oracledbconfig, **function**(err, connection) {

**if** (err) {

console.error('Error connecting to Oracle SQL DB - ' + err);

**return**;

}

console.log('Oracle DB Connection established');

connection.execute(

"select \* from BANK\_TRANS\_ARTIVA where BANK\_TRANS\_ID = :id",

[ 3811 ], // bind value for :id

**function**(err, result) {

**if** (err) {

console.error(err.message);

**return**;

}

console.log('Select All BANK\_TRANS\_ARTIVA data');

console.log(result.rows);

});

});

1. **MS SQL Server – mssql**

Please execute below command to install MS sql DB connector.

**npm install --save mssql**

Example program to connect DB using mssql connector:-

**var** mssql = require('mssql');

**var** mssqldbconfig = {

server : "10.10.50.35",

user : "sa",

password : "Asta1234",

database : "esgqlogdb",

port : 1433

};

**var** mssqlConnection = **new** mssql.Connection(dbSettings.mssqldbconfig);

**var** request = **new** mssql.Request(mssqlConnection);

mssqlConnection.connect(**function**(err) {

**if** (err) {

console.log('Error connecting to MS SQL DB - ' + err);

**return**;

} **else** {

console.log('MS SQL DB Connection established');

request.query('select \* from dbo.category where category in (1,2,3,4)',

**function**(err, result) {

**if** (err) {

console.error(err);

} **else** {

console.log('Select All category log');

console.log(result);

}

mssqlConnection.close();

});

}

});

1. Mongo DB - mongoose

Please execute below command to install Mongo DB connector.

**npm install --save mongoose**

Example program to connect DB using mongoose connector:-

// grab the things we need

var mongoose = require('mongoose');

// Mongoose connection to MongoDB (ted/ted is readonly)

mongoose.connect(' mongodb://10.10.50.200:61797/AstaIT', function (error) {

if (error) {

console.log(error);

}

});

var Schema = mongoose.Schema;

// create a schema

var userSchema = new Schema({

name: String,

username: { type: String, required: true, unique: true },

password: { type: String, required: true },

admin: Boolean,

location: String,

meta: {

age: Number,

website: String

},

created\_at: Date,

updated\_at: Date

});

// the schema is useless so far

// we need to create a model using it

var User = mongoose.model('User', userSchema);

// make this available to our users in our Node applications

module.exports = User;

// get all the users

User.find({}, function(err, users) {

if (err) throw err;

// object of all the users

console.log(users);

});

// get the user starlord55

User.find({ username: 'starlord55' }, function(err, user) {

if (err) throw err;

// object of the user

console.log(user);

});

// get a user with ID of 1

User.findById(1, function(err, user) {

if (err) throw err;

// show the one user

console.log(user);

});

// get any admin that was created in the past month

// get the date 1 month ago

var monthAgo = new Date();

monthAgo.setMonth(monthAgo.getMonth() - 1);

User.find({ admin: true }).where('created\_at').gt(monthAgo).exec(function(err, users) {

if (err) throw err;

// show the admins in the past month

console.log(users);

});

**Ref: https://scotch.io/tutorials/using-mongoosejs-in-node-js-and-mongodb-applications**