**Node js Express Insert Data From Form into MySQL Database**

Store data from html form into MySQL database in node js express app; In this tutorial, you will learn how to insert/save/store data from HTML bootstrap form into database using node js express app.

This tutorial will create a simple contact form using bootstrap 4 library and then create route in server.js file for inserting data into MySQL database in node js express app.

And as well you will learn how to send data from html form node js express route and store form data into mysql database using node js express app.

**How to Insert/Save Data From Form into MySQL Database in Node js Express**

* Step 1 – Create Node Express js App
* Step 2 – Create Table in MySQL Database
* Step 3 – Install express flash ejs body-parser mysql dependencies
* Step 4 – Create HTML Markup Form
* Step 5 – Create Server.js File and Database Connection File
* Step 6 – Start App Server

**Step 1 – Create Node Express js App**

Execute the following command on terminal to create node js app:

mkdir my-app

cd my-app

npm init -y

**Step 2 – Create Table in MySQL Database**

Execute the following sql query to create a table into your database:

CREATE TABLE `contacts` (

`id` int(11) NOT NULL,

`f\_name` varchar(100) NOT NULL,

`l\_name` varchar(100) NOT NULL,

`email` varchar(100) NOT NULL,

`message` text NOT NULL,

`created\_at` timestamp NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Recommended:-**[Node Js Google ReCaptcha v3 Example](https://www.tutsmake.com/node-js-google-recaptcha-v3-example/)

**Step 3 – Install express flash ejs body-parser mysql Modules**

Execute the following command on the terminal to express flash ejs body-parser mysql dependencies :

npm install -g express-generator

npx express --view=ejs

npm install

npm install express-flash --save

npm install express-session --save

npm install body-parser --save

npm install mysql --save

[body-parser](https://www.npmjs.com/package/body-parser) – Node.js request body parsing middleware which parses the incoming request body before your handlers, and make it available under **req.body** property. In other words, it simplifies the incoming request.

[Express-Flash](https://www.npmjs.com/package/express-flash) – Flash Messages for your Express Application. Flash is **an extension of connect-flash with the ability to define a flash message and render it without** redirecting the request.

[Express-Session](https://www.npmjs.com/package/express-session)– Express-session – **an HTTP server-side framework used to create and manage a session middleware**.

[Express-EJS](https://www.npmjs.com/package/express-ejs-layouts)– EJS is **a simple templating language** which is used to generate HTML markup with plain JavaScript. It also helps to embed JavaScript to HTML pages

[Mysql](https://www.mysql.com/)– **an open-source relational database management system (RDBMS)**.

**Recommended:-**[Node js Express MySQL User Authentication Rest API Example](https://www.tutsmake.com/node-js-express-mysql-user-authentication-rest-api-example/)

**Step 4 – Create HTML Markup Form**

Create html form for inserting data into database; So visit **views** directory and create index.ejs file inside it. Then add the following code into it:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | <!DOCTYPE html>  <html lang="en">    <head>      <title>Node.js Express Save Data from Html Form to Mysql Database - Tutsmake.com</title>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1">      <link rel="stylesheet" href="<https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css>">    </head>    <body>      <div class="container mt-4">        <div class="card">          <div class="card-body">            <% if (messages.success) { %>                <p class="alert alert-success m2-4 mb-2"><%- messages.success %></p>            <% } %>              <h2>Node.js Express Save Data from Html Form to Mysql Database - Tutsmake.com</h2>              <form action="contact-us" method="POST">              <div class="form-group">                <label for="firstName">First Name</label>                <input type="text" class="form-control col-lg-9" id="f\_name" aria-describedby="emailHelp" placeholder="Enter first name" name="f\_name">              </div>              <div class="form-group">                <label for="lastName">Last Name</label>                <input type="text" class="form-control col-lg-9" id="l\_name" aria-describedby="emailHelp" placeholder="Enter last name" name="l\_name">              </div>              <div class="form-group">                <label for="exampleInputEmail1">Email address</label>                <input type="email" class="form-control col-lg-9" id="exampleInputEmail1" aria-describedby="emailHelp" name="email" placeholder="Enter email">              </div>              <div class="form-group">                <label for="exampleInputEmail1">Message</label>                <textarea name="message" class="form-control col-lg-9"></textarea>              </div>              <button type="submit" class="btn btn-primary">Submit</button>            </form>      </div>    </div>    </body>  </html> |

**Recommended:-**[Angular 12 CRUD + Node.js + Express + MySQL Tutorial](https://www.tutsmake.com/angular-12-crud-node-js-express-mysql-tutorial/)

**Step 5 – Create Server.js and Database Connection File**

Create server.js file and import express flash session body-parser mysql dependencies in server.js; as shown below:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73 | var createError = require('http-errors');  var express = require('express');  var path = require('path');  var cookieParser = require('cookie-parser');  var logger = require('morgan');  var bodyParser = require('body-parser');  var flash = require('express-flash');  var session = require('express-session');  var db=require('./database');    var app = express();    // view engine setup  app.set('views', path.join(\_\_dirname, 'views'));  app.set('view engine', 'ejs');    app.use(logger('dev'));  app.use(express.json());  app.use(express.urlencoded({ extended: false }));  app.use(cookieParser());  app.use(express.static(path.join(\_\_dirname, 'public')));    app.use(session({      secret: '123456catr',      resave: false,      saveUninitialized: true,      cookie: { maxAge: 60000 }  }))    app.use(flash());    /\* GET home page. \*/  app.get('/', function(req, res, next) {    res.render('contact-us', { title: 'Contact-Us' });  });    app.post('/contact-us', function(req, res, next) {    var f\_name = req.body.f\_name;    var l\_name = req.body.l\_name;    var email = req.body.email;    var message = req.body.message;      var sql = `INSERT INTO contacts (f\_name, l\_name, email, message, created\_at) VALUES ("${f\_name}", "${l\_name}", "${email}", "${message}", NOW())`;    db.query(sql, function(err, result) {      if (err) throw err;      console.log('record inserted');      req.flash('success', 'Data added successfully!');      res.redirect('/');    });  });    // catch 404 and forward to error handler  app.use(function(req, res, next) {    next(createError(404));  });    // error handler  app.use(function(err, req, res, next) {    // set locals, only providing error in development    res.locals.message = err.message;    res.locals.error = req.app.get('env') === 'development' ? err : {};      // render the error page    res.status(err.status || 500);    res.render('error');  });    // port must be set to 3000 because incoming http requests are routed from port 80 to port 8080  app.listen(3000, function () {      console.log('Node app is running on port 3000');  });    module.exports = app; |

Note that; The following route in server.js will show form and store form data into MySQL database and node js experss app:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | /\* GET home page. \*/  app.get('/', function(req, res, next) {    res.render('contact-us', { title: 'Contact-Us' });  });    app.post('/contact-us', function(req, res, next) {    var f\_name = req.body.f\_name;    var l\_name = req.body.l\_name;    var email = req.body.email;    var message = req.body.message;      var sql = `INSERT INTO contacts (f\_name, l\_name, email, message, created\_at) VALUES ("${f\_name}", "${l\_name}", "${email}", "${message}", NOW())`;    db.query(sql, function(err, result) {      if (err) throw err;      console.log('record inserted');      req.flash('success', 'Data added successfully!');      res.redirect('/');    });  }); |

Create database.js file and add the following code into it to connect your app to database:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | var mysql = require('mysql');  var conn = mysql.createConnection({    host: 'localhost', // Replace with your host name    user: 'root',      // Replace with your database username    password: '',      // Replace with your database password    database: 'my-node' // // Replace with your database Name  });    conn.connect(function(err) {    if (err) throw err;    console.log('Database is connected successfully !');  });  module.exports = conn; |

**Recommended:-**[Create Rest API Using Node.js, Express and MySQL](https://www.tutsmake.com/node-express-js-creating-a-restful-api-mysql-example/)

**Step 6 – Start App Server**

You can use the following command to start node js app server:

//run the below command

npm start

after run this command open your browser and hit

http://127.0.0.1:3000/