**Index**

1. [**Download Node Express Module & Application**](https://www.webslesson.info/2022/04/insert-update-delete-data-from-mysql-in-node-js-using-express-js.html#step1)
2. [**Basic Routing of Node Express Application**](https://www.webslesson.info/2022/04/insert-update-delete-data-from-mysql-in-node-js-using-express-js.html#step2)
3. [**Make MySQL Database Connection in Node Express Application**](https://www.webslesson.info/2022/04/insert-update-delete-data-from-mysql-in-node-js-using-express-js.html#step3)
4. [**Fetch & Display Data from MySQL Database**](https://www.webslesson.info/2022/04/insert-update-delete-data-from-mysql-in-node-js-using-express-js.html#step4)
5. [**Insert Form Data into MySQL Table**](https://www.webslesson.info/2022/04/insert-update-delete-data-from-mysql-in-node-js-using-express-js.html#step5)
6. [**Edit or Update MySQL Table Data**](https://www.webslesson.info/2022/04/insert-update-delete-data-from-mysql-in-node-js-using-express-js.html#step6)
7. [**Delete Data From MySQL Table**](https://www.webslesson.info/2022/04/insert-update-delete-data-from-mysql-in-node-js-using-express-js.html#step7)
8. [**Display Flash Message using Connect-flash Module**](https://www.webslesson.info/2022/04/insert-update-delete-data-from-mysql-in-node-js-using-express-js.html#step8)

## Step 1 - Download Node Express Module & Application

For build web application in Node.js, here we will use Node Express JS application. So by using this Node Express framework we can easily and fastly developed web application in the Node.js.

Sof first we need to Install Express Generator. So for this, we have to go command prompt and create one **crud** directory and goes into that directory.

mkdir crud

cd crud

And under this directory, we have to run following command. This command will install the Express generator.

F:\node\crud> npm install -g express-generator

Next we need to install Express Application, so for this we have to run following command, which will install express application.

npx express --view=ejs

In the above command, **ejs** is a template engine of the Node.js Express and it will download Node Express Application in the define directory.

create : public\

create : public\javascripts\

create : public\images\

create : public\stylesheets\

create : public\stylesheets\style.css

create : routes\

create : routes\index.js

create : routes\users.js

create : views\

create : views\error.ejs

create : views\index.ejs

create : app.js

create : package.json

create : bin\

create : bin\www

install dependencies:

> npm install

run the app:

> SET DEBUG=crud:\* & npm start

Next we need to install dependecies, so for this we have to run following command. So this command will install dependencies in Node Express Application.

Now we want to check output in the browser, we have go to command prompt and run followng command.

F:\node\crud>npm start

This command will start Node.js server and for check output in the browser, we have need to hit following url in the browser.

http://localhost:3000

## Step 2 - Basic Routing of Node Express Application

After install application, now we want to create one javascript file in the **routes** directory. So here we have create one **sample\_data.js** file in the routes directory. And under this file, we have create simple Node Express application with simple routing for crud application.

**routes/sample\_data.js**

var express = require('express');

var router = express.Router();

router.get("/", function(request, response, next){

response.send('List all Sample Data');

});

router.get("/add", function(request, response, next){

response.send('Add Sample Data');

});

module.exports = router;

Next we have to open **app.js** file and under this file, we have need to add following code.

var sampledataRouter = require('./routes/sample\_data');

After this, for set route for the sample\_data, we have again add following code in **app.js** file.

app.use('/sample\_data', sampledataRouter);

Now our simple Node Express Application is ready for check simple routing in the browser. Now you can hit following URL in the browser for check output of Node.js Express Application in the browser.

http://localhost:3000/sample\_data

http://localhost:3000/sample\_data/add

## Step 3 - Make MySQL Database Connection in Node Express Application

If you are making any dynamic application with Datatabase, so first you need to create Database connection. So Here we have use MySQL Database with Node.js Express Application. So for make MySQL database connection from Node js Express application, So first we need to download MySQL node package and with the help of this package, it will do Insert, Update, Delete and Read Database operation in Node js express application. So for download and install MySQL node package, we need to run following command in command prompt.

npm install mysql

After download and install Node JS Express and MySQL package, now we have go to texteditor and create **database.js** file and under this file, we have to write following code in it. So it will make mysql database connection file in the Node application.

**database.js**

const mysql = require('mysql');

var connection = mysql.createConnection({

host : 'localhost',

database : 'testing',

user : 'root',

password : ''

});

connection.connect(function(error){

if(error)

{

throw error;

}

else

{

console.log('MySQL Database is connected Successfully');

}

});

module.exports = connection;

After creating database connection file, next we need to include that file in javascript server file, so here below we have include database connection file under **routes/sample\_data.js** file, and after including database connection file, so that file has been connected with mysql database and it can do mysql database operation.

**routes/sample\_data.js**

var express = require('express');

var router = express.Router();

var database = require('../database');

router.get("/", function(request, response, next){

response.send('List all Sample Data');

});

router.get("/add", function(request, response, next){

response.send('Add Sample Data');

});

module.exports = router;

For Create MySQL table in your MySQL Database, you have to run followng query, which will create **sample\_data** table in your MySQL Database.

--

-- Table structure for table `sample\_data`

--

CREATE TABLE `sample\_data` (

`id` int(10) NOT NULL,

`first\_name` varchar(250) NOT NULL,

`last\_name` varchar(250) NOT NULL,

`age` varchar(30) NOT NULL,

`gender` varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `sample\_data`

--

INSERT INTO `sample\_data` (`id`, `first\_name`, `last\_name`, `age`, `gender`) VALUES

(1, 'John', 'Smith', '26', 'Male'),

(2, 'Donna', 'Hubber', '24', 'Female'),

(3, 'Peter', 'Parker', '28', 'Male'),

(4, 'Tom', 'Muddy', '32', 'Male'),

(6, 'Lisa', 'Ray', '26', 'Female');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `sample\_data`

--

ALTER TABLE `sample\_data`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `sample\_data`

--

ALTER TABLE `sample\_data`

MODIFY `id` int(10) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=7;

## Step 4 - Fetch & Display Data from MySQL Database

In this step, you can learn How to fetch data from MySQL Table and display on the web page in the HTML table format in the Node.js application.

**routes/sample\_data.js**

var express = require('express');

var router = express.Router();

var database = require('../database');

router.get("/", function(request, response, next){

var query = "SELECT \* FROM sample\_data ORDER BY id DESC";

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.render('sample\_data', {title:'Node.js MySQL CRUD Application', action:'list', sampleData:data});

}

});

});

module.exports = router;

So for fetch data from database, first you have to create route in the route javascript server file using get() function and under this, you have to write MySQL query for fetch data from MySQL table.

Under this Route function, you have to execute Mysql Fetch query by using query() function, so this function will execute MySQL fetch query and return query exection result which we want to send to node.js template file.

Next we want to load node js template file, so by using response.render() function, it will load node js template file, This function has two parameter, in first parameter, you have to define template file name, and in second paramter, you have to pass data which can be access in template file, and here you have to define data in json data format.

After this you have go to node.js template file, so here we have create template file in **views/sample\_data.ejs**. This file will be load in the browser by using get() function. Here we will use sample file for different operation by using if condition. Below you can find source code for how display mysql table data in HTML format using Nodejs.

**views/sample\_data.ejs**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title></title>

</head>

<body>

<div class="container">

<h1 class="text-center mt-3 mb-3"><%= title %></h1>

<% if(action == 'add') { %>

<% } else { %>

<div class="card">

<div class="card-header">

<div class="row">

<div class="col">Sample Data</div>

<div class="col"></div>

</div>

</div>

<div class="card-body">

<div class="table-responsive">

<table class="table table-bordered">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

<th>Gender</th>

<th>Action</th>

</tr>

<%

if(sampleData.length > 0)

{

sampleData.forEach(function(data){

%>

<tr>

<td><%= data.first\_name %></td>

<td><%= data.last\_name %></td>

<td><%= data.age %></td>

<td><%= data.gender %></td>

<td></td>

</tr>

<%

});

}

else

{

%>

<tr>

<td colspan="5">No Data Found</td>

</tr>

<%

}

%>

</table>

</div>

</div>

</div>

<% } %>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

For check Fetch and Display output in the browser, you have to hit following url in the browser, which will show you output of Node.js Fetch and Display Data from MySQL Database.

http://localhost:3000/sample\_data

## Step 5 - Insert Form Data into MySQL Table

In this Node JS CRUD Application, now we have proceed for Insert Form Data into MySQL table. Under this step you can learn How to Insert Data into MySQL Database using Node JS Express Application.

So For Insert Data into Database, first you need to create one HTML Form for get Data from user.

So for load HTML form in the browser, first we need to set route for load HTML form in the browser. So you can find source code for set route for load Add Form in the browser.

**routes/sample\_data.js**

var express = require('express');

var router = express.Router();

var database = require('../database');

router.get("/", function(request, response, next){

var query = "SELECT \* FROM sample\_data ORDER BY id DESC";

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.render('sample\_data', {title:'Node.js MySQL CRUD Application', action:'list', sampleData:data});

}

});

});

router.get("/add", function(request, response, next){

response.render("sample\_data", {title:'Insert Data into MySQL', action:'add'});

});

module.exports = router;

Once you have set route, next you have to create HTML Form in Node Express Template file, so HTML form code you can find below.

**views/sample\_data.ejs**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title></title>

</head>

<body>

<div class="container">

<h1 class="text-center mt-3 mb-3"><%= title %></h1>

<% if(action == 'add') { %>

<div class="card">

<div class="card-header">Sample Form</div>

<div class="card-body">

<form method="POST" action="/sample\_data/add\_sample\_data">

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>First Name</label>

<input type="text" name="first\_name" id="first\_name" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Last Name</label>

<input type="text" name="last\_name" id="last\_name" class="form-control" />

</div>

</div>

</div>

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>Age</label>

<input type="number" name="age" id="age" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Gender</label>

<select name="gender" class="form-control">

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

</div>

</div>

<div class="mb-3">

<input type="submit" name="submit\_button" class="btn btn-primary" value="Add" />

</div>

</form>

</div>

</div>

<% } else { %>

<div class="card">

<div class="card-header">

<div class="row">

<div class="col">Sample Data</div>

<div class="col">

<a href="/sample\_data/add" class="btn btn-success btn-sm float-end">Add</a>

</div>

</div>

</div>

<div class="card-body">

<div class="table-responsive">

<table class="table table-bordered">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

<th>Gender</th>

<th>Action</th>

</tr>

<%

if(sampleData.length > 0)

{

sampleData.forEach(function(data){

%>

<tr>

<td><%= data.first\_name %></td>

<td><%= data.last\_name %></td>

<td><%= data.age %></td>

<td><%= data.gender %></td>

<td></td>

</tr>

<%

});

}

else

{

%>

<tr>

<td colspan="5">No Data Found</td>

</tr>

<%

}

%>

</table>

</div>

</div>

</div>

<% } %>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

After creating HTML form, so when this form has been submitted, then it will send request to **add\_sample\_data** route. So now we need to create route for handle form data request.

So we have create route in **routes/sample\_data.js** and under this first it will save form data into local variable.

Then after we have to create MySQL Insert query and under that query we have to define variable in which we have store form data.

Next we have to run MySQL Insert query and it will insert form data into MySQL table in the Node JS application and it will request web page to root url in which you can see sample\_data table data with last inserted data.

**routes/sample\_data.js**

var express = require('express');

var router = express.Router();

var database = require('../database');

router.get("/", function(request, response, next){

var query = "SELECT \* FROM sample\_data ORDER BY id DESC";

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.render('sample\_data', {title:'Node.js MySQL CRUD Application', action:'list', sampleData:data});

}

});

});

router.get("/add", function(request, response, next){

response.render("sample\_data", {title:'Insert Data into MySQL', action:'add'});

});

router.post("/add\_sample\_data", function(request, response, next){

var first\_name = request.body.first\_name;

var last\_name = request.body.last\_name;

var age = request.body.age;

var gender = request.body.gender;

var query = `

INSERT INTO sample\_data

(first\_name, last\_name, age, gender)

VALUES ("${first\_name}", "${last\_name}", "${age}", "${gender}")

`;

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.redirect("/sample\_data");

}

});

});

module.exports = router;

For check output of Insert Form Data into MySQL using Node JS in the browser, first you need to start Node.js server from command prompt by run **npm start** command and then after go to browser and hit following url.

http://localhost:3000/sample\_data/add

## Step 6 - Edit or Update MySQL Table Data

Once we have insert form data into MySQL table using Node.js, next we need to show you how to edit or update MySQL table data in Node.js Express Application.

So first we need to create Edit button in each of MySQL data in HTML table. So we have go to **views/sample\_data.ejs** file and under this you have to write following code for create edit button.

**views/sample\_data.ejs**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title></title>

</head>

<body>

<div class="container">

<h1 class="text-center mt-3 mb-3"><%= title %></h1>

<% if(action == 'add') { %>

<div class="card">

<div class="card-header">Sample Form</div>

<div class="card-body">

<form method="POST" action="/sample\_data/add\_sample\_data">

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>First Name</label>

<input type="text" name="first\_name" id="first\_name" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Last Name</label>

<input type="text" name="last\_name" id="last\_name" class="form-control" />

</div>

</div>

</div>

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>Age</label>

<input type="number" name="age" id="age" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Gender</label>

<select name="gender" class="form-control">

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

</div>

</div>

<div class="mb-3">

<input type="submit" name="submit\_button" class="btn btn-primary" value="Add" />

</div>

</form>

</div>

</div>

<% } else { %>

<div class="card">

<div class="card-header">

<div class="row">

<div class="col">Sample Data</div>

<div class="col">

<a href="/sample\_data/add" class="btn btn-success btn-sm float-end">Add</a>

</div>

</div>

</div>

<div class="card-body">

<div class="table-responsive">

<table class="table table-bordered">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

<th>Gender</th>

<th>Action</th>

</tr>

<%

if(sampleData.length > 0)

{

sampleData.forEach(function(data){

%>

<tr>

<td><%= data.first\_name %></td>

<td><%= data.last\_name %></td>

<td><%= data.age %></td>

<td><%= data.gender %></td>

<td>

<a href="/sample\_data/edit/<%= data.id %>" class="btn btn-primary btn-sm">Edit</a>

</td>

</tr>

<%

});

}

else

{

%>

<tr>

<td colspan="5">No Data Found</td>

</tr>

<%

}

%>

</table>

</div>

</div>

</div>

<% } %>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

After this, we have go to **routes/sample\_data.js** and under this file, we have to create new route for fetch single row of data, and then load that data into HTML form and load that edit data form in the browser.

**routes/sample\_data.js**

router.get('/edit/:id', function(request, response, next){

var id = request.params.id;

var query = `SELECT \* FROM sample\_data WHERE id = "${id}"`;

database.query(query, function(error, data){

response.render('sample\_data', {title: 'Edit MySQL Table Data', action:'edit', sampleData:data[0]});

});

});

Under this routes, we have first MySQL select query for fetch single row of data, and execute that query and it has fetch data from MySQL table, After this we have send that data to Node js template file, which you can seen above.

After this, again we have go to **views/sample\_data.ejs** template file, and under this file, first we have write if condition for load edit form in browser with data.

**views/sample\_data.ejs**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title></title>

</head>

<body>

<div class="container">

<h1 class="text-center mt-3 mb-3"><%= title %></h1>

<% if(action == 'add') { %>

<div class="card">

<div class="card-header">Sample Form</div>

<div class="card-body">

<form method="POST" action="/sample\_data/add\_sample\_data">

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>First Name</label>

<input type="text" name="first\_name" id="first\_name" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Last Name</label>

<input type="text" name="last\_name" id="last\_name" class="form-control" />

</div>

</div>

</div>

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>Age</label>

<input type="number" name="age" id="age" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Gender</label>

<select name="gender" class="form-control">

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

</div>

</div>

<div class="mb-3">

<input type="submit" name="submit\_button" class="btn btn-primary" value="Add" />

</div>

</form>

</div>

</div>

<% } else if(action == 'edit') { %>

<div class="card">

<div class="card-header">Sample Form</div>

<div class="card-body">

<form method="POST" action="/sample\_data/edit/<%= sampleData.id %>">

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>First Name</label>

<input type="text" name="first\_name" id="first\_name" class="form-control" value="<%= sampleData.first\_name %>" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Last Name</label>

<input type="text" name="last\_name" id="last\_name" class="form-control" value="<%= sampleData.last\_name %>" />

</div>

</div>

</div>

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>Age</label>

<input type="number" name="age" id="age" class="form-control" value="<%= sampleData.age %>" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Gender</label>

<select name="gender" id="gender" class="form-control">

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

</div>

</div>

<div class="mb-3">

<input type="submit" name="submit\_button" class="btn btn-primary" value="Edit" />

</div>

</form>

<script>

document.getElementById('gender').value="<%= sampleData.gender %>";

</script>

</div>

</div>

<% } else { %>

<div class="card">

<div class="card-header">

<div class="row">

<div class="col">Sample Data</div>

<div class="col">

<a href="/sample\_data/add" class="btn btn-success btn-sm float-end">Add</a>

</div>

</div>

</div>

<div class="card-body">

<div class="table-responsive">

<table class="table table-bordered">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

<th>Gender</th>

<th>Action</th>

</tr>

<%

if(sampleData.length > 0)

{

sampleData.forEach(function(data){

%>

<tr>

<td><%= data.first\_name %></td>

<td><%= data.last\_name %></td>

<td><%= data.age %></td>

<td><%= data.gender %></td>

<td>

<a href="/sample\_data/edit/<%= data.id %>" class="btn btn-primary btn-sm">Edit</a>

</td>

</tr>

<%

});

}

else

{

%>

<tr>

<td colspan="5">No Data Found</td>

</tr>

<%

}

%>

</table>

</div>

</div>

</div>

<% } %>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

Next we have again go to **routes/sample\_data.js** file and under this file, we have again create routes for handle edit form data request for edit data. Under this routes, we have execute update data query and after update data it will redirect to web page on which we have display mysql data in html table format.

**routes/sample\_data.js**

var express = require('express');

var router = express.Router();

var database = require('../database');

router.get("/", function(request, response, next){

var query = "SELECT \* FROM sample\_data ORDER BY id DESC";

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.render('sample\_data', {title:'Node.js MySQL CRUD Application', action:'list', sampleData:data});

}

});

});

router.get("/add", function(request, response, next){

response.render("sample\_data", {title:'Insert Data into MySQL', action:'add'});

});

router.post("/add\_sample\_data", function(request, response, next){

var first\_name = request.body.first\_name;

var last\_name = request.body.last\_name;

var age = request.body.age;

var gender = request.body.gender;

var query = `

INSERT INTO sample\_data

(first\_name, last\_name, age, gender)

VALUES ("${first\_name}", "${last\_name}", "${age}", "${gender}")

`;

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.redirect("/sample\_data");

}

});

});

router.get('/edit/:id', function(request, response, next){

var id = request.params.id;

var query = `SELECT \* FROM sample\_data WHERE id = "${id}"`;

database.query(query, function(error, data){

response.render('sample\_data', {title: 'Edit MySQL Table Data', action:'edit', sampleData:data[0]});

});

});

router.post('/edit/:id', function(request, response, next){

var id = request.params.id;

var first\_name = request.body.first\_name;

var last\_name = request.body.last\_name;

var age = request.body.age;

var gender = request.body.gender;

var query = `

UPDATE sample\_data

SET first\_name = "${first\_name}",

last\_name = "${last\_name}",

age = "${age}",

gender = "${gender}"

WHERE id = "${id}"

`;

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.redirect('/sample\_data');

}

});

});

module.exports = router;

**Step 7 - Delete Data From MySQL Table**

Under this step, you can find how to Delete or Remove Data from MySQL Database in Node.js Express Application.

For Delete or Remove first you need to create Delete button in each row of data. So you have to open **views/sample\_data.ejs** file and create delete button in display MySQL data in HTML table format.

**views/sample\_data.ejs**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title></title>

</head>

<body>

<div class="container">

<h1 class="text-center mt-3 mb-3"><%= title %></h1>

<% if(action == 'add') { %>

<div class="card">

<div class="card-header">Sample Form</div>

<div class="card-body">

<form method="POST" action="/sample\_data/add\_sample\_data">

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>First Name</label>

<input type="text" name="first\_name" id="first\_name" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Last Name</label>

<input type="text" name="last\_name" id="last\_name" class="form-control" />

</div>

</div>

</div>

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>Age</label>

<input type="number" name="age" id="age" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Gender</label>

<select name="gender" class="form-control">

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

</div>

</div>

<div class="mb-3">

<input type="submit" name="submit\_button" class="btn btn-primary" value="Add" />

</div>

</form>

</div>

</div>

<% } else if(action == 'edit') { %>

<div class="card">

<div class="card-header">Sample Form</div>

<div class="card-body">

<form method="POST" action="/sample\_data/edit/<%= sampleData.id %>">

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>First Name</label>

<input type="text" name="first\_name" id="first\_name" class="form-control" value="<%= sampleData.first\_name %>" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Last Name</label>

<input type="text" name="last\_name" id="last\_name" class="form-control" value="<%= sampleData.last\_name %>" />

</div>

</div>

</div>

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>Age</label>

<input type="number" name="age" id="age" class="form-control" value="<%= sampleData.age %>" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Gender</label>

<select name="gender" id="gender" class="form-control">

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

</div>

</div>

<div class="mb-3">

<input type="submit" name="submit\_button" class="btn btn-primary" value="Edit" />

</div>

</form>

<script>

document.getElementById('gender').value="<%= sampleData.gender %>";

</script>

</div>

</div>

<% } else { %>

<div class="card">

<div class="card-header">

<div class="row">

<div class="col">Sample Data</div>

<div class="col">

<a href="/sample\_data/add" class="btn btn-success btn-sm float-end">Add</a>

</div>

</div>

</div>

<div class="card-body">

<div class="table-responsive">

<table class="table table-bordered">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

<th>Gender</th>

<th>Action</th>

</tr>

<%

if(sampleData.length > 0)

{

sampleData.forEach(function(data){

%>

<tr>

<td><%= data.first\_name %></td>

<td><%= data.last\_name %></td>

<td><%= data.age %></td>

<td><%= data.gender %></td>

<td>

<a href="/sample\_data/edit/<%= data.id %>" class="btn btn-primary btn-sm">Edit</a>

<a href="/sample\_data/delete/<%= data.id %>" class="btn btn-danger btn-sm">Delete</a>

</td>

</tr>

<%

});

}

else

{

%>

<tr>

<td colspan="5">No Data Found</td>

</tr>

<%

}

%>

</table>

</div>

</div>

</div>

<% } %>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

After creating delete button in each row of data, next we need to create route for handle delete data request. So under that route, we have to create one delete data query which will get primary key value from URL and based on that value it will delete data from MySQL table and it will redirect web page to another url which will display remaining data on the web page in HTML table format.

**routes/sample\_data.js**

var express = require('express');

var router = express.Router();

var database = require('../database');

router.get("/", function(request, response, next){

var query = "SELECT \* FROM sample\_data ORDER BY id DESC";

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.render('sample\_data', {title:'Node.js MySQL CRUD Application', action:'list', sampleData:data});

}

});

});

router.get("/add", function(request, response, next){

response.render("sample\_data", {title:'Insert Data into MySQL', action:'add'});

});

router.post("/add\_sample\_data", function(request, response, next){

var first\_name = request.body.first\_name;

var last\_name = request.body.last\_name;

var age = request.body.age;

var gender = request.body.gender;

var query = `

INSERT INTO sample\_data

(first\_name, last\_name, age, gender)

VALUES ("${first\_name}", "${last\_name}", "${age}", "${gender}")

`;

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.redirect("/sample\_data");

}

});

});

router.get('/edit/:id', function(request, response, next){

var id = request.params.id;

var query = `SELECT \* FROM sample\_data WHERE id = "${id}"`;

database.query(query, function(error, data){

response.render('sample\_data', {title: 'Edit MySQL Table Data', action:'edit', sampleData:data[0]});

});

});

router.post('/edit/:id', function(request, response, next){

var id = request.params.id;

var first\_name = request.body.first\_name;

var last\_name = request.body.last\_name;

var age = request.body.age;

var gender = request.body.gender;

var query = `

UPDATE sample\_data

SET first\_name = "${first\_name}",

last\_name = "${last\_name}",

age = "${age}",

gender = "${gender}"

WHERE id = "${id}"

`;

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.redirect('/sample\_data');

}

});

});

router.get('/delete/:id', function(request, response, next){

var id = request.params.id;

var query = `

DELETE FROM sample\_data WHERE id = "${id}"

`;

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.redirect("/sample\_data");

}

});

});

module.exports = router;

## Step 8 - Display Flash Message using Connect-flash Module

After creating CRUD Application in Node.js, now we need to display success message on CRUD (Create, Read, Update, Delete) operation. So for this, here in this step we will show you how to display Flash Message in Node JS by using Connect-flash Module.

So first we need to download **express-session** and **connect-flash** module under node application. So for this you go into command prompt and run following command. So this command will download and install express-session and connect-flash module in your node application.

npm install connect-flash express express-session --save

Next you have to import this module into your Node application, so for this, you have to open **app.js** file.

var session = require('express-session');

var flash = require('connect-flash');

After this you have implement Connect flash module and express session module, so for this, you have to define following configruation in your **app.js** file.

app.use(session({

secret:'webslesson',

cookie: {maxAge : 60000},

saveUninitialized: false,

resave: false

}));

app.use(flash());

After this, for add flash message in routes server file, you have to open **routes/sample\_data.php** file and under this file, you have to add flash message on each operation.

**routes/sample\_data.js**

var express = require('express');

var router = express.Router();

var database = require('../database');

router.get("/", function(request, response, next){

var query = "SELECT \* FROM sample\_data ORDER BY id DESC";

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

response.render('sample\_data', {title:'Node.js MySQL CRUD Application', action:'list', sampleData:data, message:request.flash('success')});

}

});

});

router.get("/add", function(request, response, next){

response.render("sample\_data", {title:'Insert Data into MySQL', action:'add'});

});

router.post("/add\_sample\_data", function(request, response, next){

var first\_name = request.body.first\_name;

var last\_name = request.body.last\_name;

var age = request.body.age;

var gender = request.body.gender;

var query = `

INSERT INTO sample\_data

(first\_name, last\_name, age, gender)

VALUES ("${first\_name}", "${last\_name}", "${age}", "${gender}")

`;

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

request.flash('success', 'Sample Data Inserted');

response.redirect("/sample\_data");

}

});

});

router.get('/edit/:id', function(request, response, next){

var id = request.params.id;

var query = `SELECT \* FROM sample\_data WHERE id = "${id}"`;

database.query(query, function(error, data){

response.render('sample\_data', {title: 'Edit MySQL Table Data', action:'edit', sampleData:data[0]});

});

});

router.post('/edit/:id', function(request, response, next){

var id = request.params.id;

var first\_name = request.body.first\_name;

var last\_name = request.body.last\_name;

var age = request.body.age;

var gender = request.body.gender;

var query = `

UPDATE sample\_data

SET first\_name = "${first\_name}",

last\_name = "${last\_name}",

age = "${age}",

gender = "${gender}"

WHERE id = "${id}"

`;

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

request.flash('success', 'Sample Data Updated');

response.redirect('/sample\_data');

}

});

});

router.get('/delete/:id', function(request, response, next){

var id = request.params.id;

var query = `

DELETE FROM sample\_data WHERE id = "${id}"

`;

database.query(query, function(error, data){

if(error)

{

throw error;

}

else

{

request.flash('success', 'Sample Data Deleted');

response.redirect("/sample\_data");

}

});

});

module.exports = router;

Lastly you have to display flash message in template file, so you have to open **views/sample\_data.ejs** file and under this you have to define flash message.

**views/sample\_data.ejs**

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title></title>

</head>

<body>

<div class="container">

<h1 class="text-center mt-3 mb-3"><%= title %></h1>

<% if(action == 'add') { %>

<div class="card">

<div class="card-header">Sample Form</div>

<div class="card-body">

<form method="POST" action="/sample\_data/add\_sample\_data">

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>First Name</label>

<input type="text" name="first\_name" id="first\_name" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Last Name</label>

<input type="text" name="last\_name" id="last\_name" class="form-control" />

</div>

</div>

</div>

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>Age</label>

<input type="number" name="age" id="age" class="form-control" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Gender</label>

<select name="gender" class="form-control">

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

</div>

</div>

<div class="mb-3">

<input type="submit" name="submit\_button" class="btn btn-primary" value="Add" />

</div>

</form>

</div>

</div>

<% } else if(action == 'edit') { %>

<div class="card">

<div class="card-header">Sample Form</div>

<div class="card-body">

<form method="POST" action="/sample\_data/edit/<%= sampleData.id %>">

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>First Name</label>

<input type="text" name="first\_name" id="first\_name" class="form-control" value="<%= sampleData.first\_name %>" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Last Name</label>

<input type="text" name="last\_name" id="last\_name" class="form-control" value="<%= sampleData.last\_name %>" />

</div>

</div>

</div>

<div class="row">

<div class="col-md-6">

<div class="mb-3">

<label>Age</label>

<input type="number" name="age" id="age" class="form-control" value="<%= sampleData.age %>" />

</div>

</div>

<div class="col-md-6">

<div class="mb-3">

<label>Gender</label>

<select name="gender" id="gender" class="form-control">

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

</div>

</div>

<div class="mb-3">

<input type="submit" name="submit\_button" class="btn btn-primary" value="Edit" />

</div>

</form>

<script>

document.getElementById('gender').value="<%= sampleData.gender %>";

</script>

</div>

</div>

<% } else { %>

<% if(message.length > 0) { %>

<div class="alert alert-success">

<%= message %>

</div>

<% } %>

<div class="card">

<div class="card-header">

<div class="row">

<div class="col">Sample Data</div>

<div class="col">

<a href="/sample\_data/add" class="btn btn-success btn-sm float-end">Add</a>

</div>

</div>

</div>

<div class="card-body">

<div class="table-responsive">

<table class="table table-bordered">

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Age</th>

<th>Gender</th>

<th>Action</th>

</tr>

<%

if(sampleData.length > 0)

{

sampleData.forEach(function(data){

%>

<tr>

<td><%= data.first\_name %></td>

<td><%= data.last\_name %></td>

<td><%= data.age %></td>

<td><%= data.gender %></td>

<td>

<a href="/sample\_data/edit/<%= data.id %>" class="btn btn-primary btn-sm">Edit</a>

<a href="/sample\_data/delete/<%= data.id %>" class="btn btn-danger btn-sm">Delete</a>

</td>

</tr>

<%

});

}

else

{

%>

<tr>

<td colspan="5">No Data Found</td>

</tr>

<%

}

%>

</table>

</div>

</div>

</div>

<% } %>

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

</body>

</html>

Now your Node JS Crud Application is ready, and you can check Node JS Insert Update Delete Application in browser, so need to go command prompt and run following command.

npm start

This command will start Node.js server and after this you have go to browser and hit following URL, so you can test your Node JS CRUD application output in the browser.

http://localhost:3000/sample\_data

So here we have stop this Node JS CRUD application and if you are beginner then please follow above step one by one and here we have also put Video tutorial also. So please watch it also you have any doubt. Hope you can build CRUD application Node js with MySQL Database.