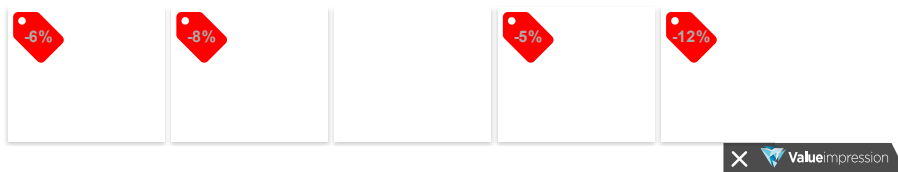


First CRUD Node Express Js Mysql Example

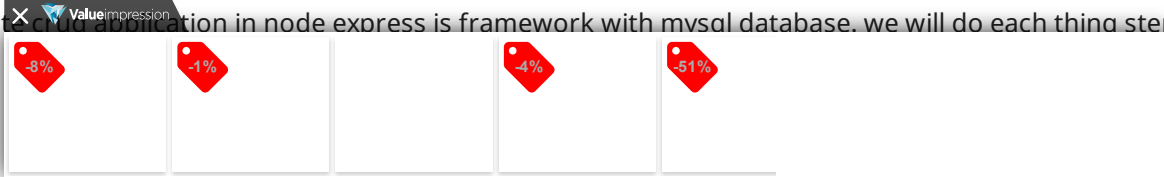


March 1, 2019 | By Tuts Make | 2 Comments

Hello developers, Today we will discuss how to install node js framework (update, read, delete) application in node js using express js framework. I will show you how to create crud application in node js.

TEL

We will create a crud application in node express is framework with mysql database. we will do each thing step by step and easy.



In this node express js crud application. We will use ejs templating engine. It is very simple and easy to understand for everyone.

Let's start create CRUD with Node.Js , Express, MySQL

Create Express js Project

Use the below command and create your express project with name expressfirst

```
express --view=ejs expressfirst
```

After successfully created expressfirst folder in your system. Next follow the below commands and install node js in your project :



```
cd expressfirst
```

```
npm install
```

Next we need



```
npm install express-flash --save
npm install express-session --save
npm install express-validator --save
npm install method-override --save
npm install mysql --save
```

express-flash

Flash is an extension of connect-flash with the ability to define a flash message and render it without redirecting the

express requirement of express-flash.

express-validator

Express validator is used to validate form data it is easy to use. express-validator highly effective and efficient way to accelerate the creation of applications.

method-override

NPM is used to run a DELETE and PUT method from an HTML form. In several web browsers only support GET and POST methods.

MySQL

Driver to connect node.js with MySQL

Database Connection with Mysql

Next we need to create one folder name lib and create a new file name db.js inside this folder. We will connect node js to mysql using this file

lib/db.js

```
var mysql=require('mysql');
var connection=mysql.createConnection({
  host:'localhost',
  user:'your username',
  password:'your password',
  database:'your database name'
});
connection.connect(function(error){
  if(!error){
    console.log(error);
```

```
}else {
  console.log('Error: ' + err);
}
});
module.exports = connection;
```

Changes in app.js

We need to some changes in app.js file. go to app.js file and put some code here :

```
var createError = require('http-errors');
var express = require('express');
```

```
var flash = require('express-flash');
var session = require('express-session');
var bodyParser = require('body-parser');

var mysql = require('mysql');
var connection = require('./lib/db');

var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');
var customersRouter = require('./routes/customers');
```

```
var app = express();
```

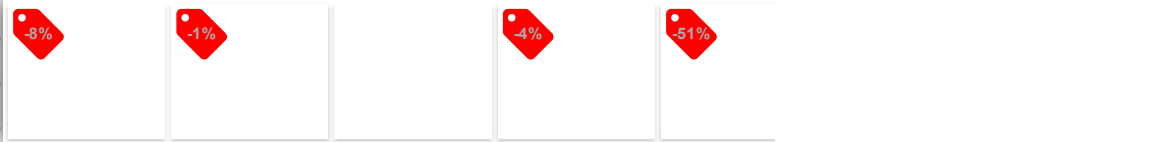
```
// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');
```

```
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
```

```
app.use(session({
  secret: '123456cat',
  resave: false,
  saveUninitialized: true,
  cookie: { maxAge: 60000 }
}));
```

```
app.use(flash());
app.use(expressValidator());
```

app.us
app.us
app.us



```
// 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;
```

```
});
module.exports = app;
```

Create Route

Next We need to create one route file name customers.js inside routes folder. After created this file, We will implement crud logic in this file. Go to routes/customers.js and use the below code :

```
1 var express = require('express');
2 var router = express.Router();
3 var connection = require('../lib/db');
4
5
6 /* GET home page. */
7 router.get('/', function(req, res, next) {
8
9   connection.query('SELECT * FROM customers ORDER BY', function(err, rows) {
10
11     if(err){
12       req.flash('error', err);
13       res.render('customers', {page_title: "Customer"});
14     }else{
15
16       res.render('customers', {page_title: "Customer"});
17     }
18   });
19
20
21 });
22
```

Valueimpression

-8%

-1%

-4%

-51%

```

24 //
25 router.post('/add', function(req, res, next){
26     req.assert('name', 'Name is required').notEmpty() //Validate name
27     req.assert('email', 'A valid email is required').isEmail() //Validate email
28
29     var errors = req.validationErrors()
30
31     if( !errors ) { //No errors were found. Passed Validation!
32
33         var user = {
34             name: req.param('name').escape().trim(),
35             email: req.param('email').escape().trim(),
36         };
37     }
38 }
39
40 // ADD NEW USER POST ACTION
41 router.post('/add', function(req, res, next){
42     req.assert('name', 'Name is required').notEmpty() //Validate name
43     req.assert('email', 'A valid email is required').isEmail() //Validate email
44
45     var errors = req.validationErrors()

```

```

53
54 // render to views/user/add.ejs
55 res.render('customers/add', {
56     title: 'Add New Customer',
57     name: user.name,
58     email: user.email
59 })
60 } else {
61     req.flash('success', 'Data added successfully!');
62     res.redirect('/customers');
63 }
64 })
65 }
66 else { //Display errors to user
67     var error_msg = ''
68     errors.forEach(function(error) {
69         error_msg += error.msg + '<br>'
70     })
71     req.flash('error', error_msg)
72
73     /**
74     * Using req.body.name
75     * because req.param('name') is deprecated
76     */
77     res.render('customers/add', {
78         title: 'Add New Customer',
79         name: req.body.name,
80         email: req.body.email
81     })
82 }
83 })
84
85 // SHOW EDIT USER FORM
86 router.get('/edit/:id', function(req, res, next){
87
88     connection.query('SELECT * FROM customers WHERE id = ' + req.params.id, function(err, rows, fi
89         if(err) throw err
90
91         // if user not found
92         if (rows.length <= 0) {
93             req.flash('error', 'Customers not found');
94             res.redirect('/customers');
95         }
96         else { // if user found
97             // render to views/user/edit.ejs template
98             res.render('customers/edit', {
99                 title: 'Edit Customer',
100                 //data: rows[0],
101                 id: rows[0].id,
102                 name: rows[0].name,
103                 email: rows[0].email
104             })

```

```

105
106
107
108 })
109
110 //
111 router.post('/update/:id', function(req, res, next) {
112     req.assert('name', 'Name is required').notEmpty() //Validate nam //Val
113     req.assert('email', 'A valid email is required').isEmail() //Validate email
114
115     var errors = req.validationErrors()
116
117     if( !errors ) {
118
119         var user = {
120             name: req.sanitize('name').escape().trim(),
121             email: req.sanitize('email').escape().trim()
122         }
123
124         connection.query('UPDATE customers SET ? WHERE id = ' + req.params.id, user, function(err, res
125             //if(err) throw err
126             if (err) {
127                 req.flash('error', err)

```

```

135         })
136     } else {
137         req.flash('success', 'Data updated successfully!');
138         res.redirect('/customers');
139     }
140 })
141
142 }
143 else { //Display errors to user
144     var error_msg = ''
145     errors.forEach(function(error) {
146         error_msg += error.msg + '<br>'
147     })
148     req.flash('error', error_msg)
149
150     /**
151     * Using req.body.name
152     * because req.param('name') is deprecated
153     */
154     res.render('customers/edit', {
155         title: 'Edit Customer',
156         id: req.params.id,
157         name: req.body.name,
158         email: req.body.email
159     })
160 }
161 })
162
163 // DELETE USER
164 router.get('/delete/:id', function(req, res, next) {
165     var user = { id: req.params.id }
166
167     connection.query('DELETE FROM customers WHERE id = ' + req.params.id, user, function(err, res)
168         //if(err) throw err
169         if (err) {
170             req.flash('error', err)
171             // redirect to users list page
172             res.redirect('/customers')
173         } else {
174             req.flash('success', 'Customer delete
175             // redirect to users list page
176             res.redirect('/customers')
177         }
178     })
179 })
180
181
182 module.exports = router;

```

First we will create a folder name customers inside the views folder.

Next we need to create the views/customers/index.ejs file.

Inside the

Create first file index.ejs

Index.ejs file, we will display the list of customers.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Customers</title>
5   <link rel='stylesheet' href='/stylesheets/style.css' />
6   <script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
7   <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity=
8   <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" rel="styl
9 </head>
10 <body>
11
12
13
14
15
16
17
18 <% } %> -->
19
20 <% if (messages.success) { %>
21   <p class="alert alert-success mt-4"><%= messages.success %></p>
22 <% } %>
23 <br>
24 <table class="table">
25 <thead>
26 <tr>
27   <th scope="col">#</th>
28   <th scope="col">Name</th>
29   <th scope="col">Email</th>
30   <th width="200px">Action</th>
31
32 </tr>
33 </thead>
34 <tbody>
35 <% if(data.length){
36
37   for(var i = 0; i< data.length; i++) {>
38     <tr>
39       <th scope="row"><%= (i+1) %></th>
40       <td><%= data[i].name%></td>
41       <td><%= data[i].email%></td>
42       <td>
43         <a class="btn btn-success edit" href="../customers/edit/<%=data[i].id%>">Edit</a>
44         <a class="btn btn-danger delete" onclick="return alert('Are You sure?')" href="../customers
45       </td>
46     </tr>
47   <% }
48
49   }else{ %>
50     <tr>
51       <td colspan="3">No user</td>
52     </tr>
53   <% } %>
54
55 </tbody>
56 </table>
57 </body>
58 </html>
```

Create second file name add.ejs

Add.ejs file, we will create form for sending to data in database.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
```


run the



npm sta

after run this command open your browser and hit

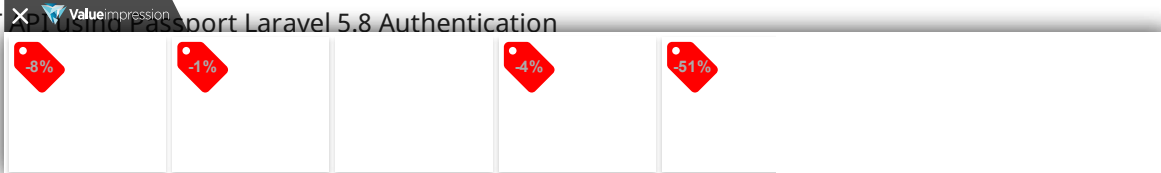
`http://127.0.0.1:3000/customers`

Conclusion

In this node express js crud tutorial – We have created node js crud (create, read, update, delete) application with mysql database. We have also successfully install node js framework express with use ejs templating engine.

2. [Laravel 6 CRUD Example Tutorial \(Step By Step\)](#)
3. [Laravel 6 Ajax CRUD\(DataTables Js\) Tutorial Example](#)
4. [First Codeigniter 3 CRUD \(Create,Read,Update,Delete\) via Mysql](#)
5. [Codeigniter 3 Create First Ajax CRUD Application](#)
6. [Node Express Js Creating a RESTful Crud Api With Mysql](#)
7. [Laravel 6 Angular JS CRUD Example Tutorial](#)

Categories: Node js



NEXT

Install Git For Windows-Mac OS X- Linux(ubuntu)

2 replies to First CRUD Node Express Js Mysql Example

Do you have any tips for using passport?
Congratulations!

Reply

Paejan
8 months ago

Hi..
Nice tutorial...
Now i wonder how to integrate this application with frontend framework like vue, Nuxt etc...

Thanks

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Valueimpression

Python Check

-8%

-1%

-4%

-51%

Python Progra

Python Program to Find the Number of Integers From 1 to n Which Contains Digits 0's

NOW
PLAYING

Categories

Angular

Best Free SEO Tools

Boostrap

Codeigniter

Composer

Git (Github)

Javascript

Jquery

Laravel

MySQL

Node js

PHP

Python

SEO

Technology

Wordpress

Recent Comments

braj on Codeigniter 4 Remove Public and Index.php From URL

Mangal Solan

-8%	-1%		-4%	-51%
-----	-----	--	-----	------

Eugene on La

w3antz on Google Recaptcha v3 Demo Example PHP

Tuts Make on Laravel PHP Simple Qr Codes Generate

Rushan on Laravel PHP Simple Qr Codes Generate

Meen on Laravel 7/6 Tutorial From Scratch | Step By Step

mahmoud on Get Country, City, latitude, longitude from IP address using PHP

Tags

8 Simple Free Seo Tools to Instantly Improve Your Marketing Today Best Free Seo Tools Create REST Ap

Laravel 6 Free Rs Chat Plugin using Nodejs Google Seo Checker **How-to-Install Larav**
to Import Export Excel in Laravel How to Install Apac

Image



How to Send Mail in Laravel [Install Node js and NPM in Windows](#) [Install PHP in Ubuntu from Scratch](#)
Laravel 5 Send Email Example [Laravel Create Update Delete Example](#) [Laravel Crud Project With MySQL](#)
[Database Laravel Form Validation Before Submit Example](#) [Laravel Import Excel to Database](#) [Laravel Import Export](#)
[Excel to Database Example](#) [Laravel Installation Process on Windows System](#) [Laravel Login Authentication Using](#)
[Email Tutorial](#) [Laravel Mail Send with Example](#) [laravel One to Many Relationship with Example](#) [Laravel Passport](#) [Laravel PHP Ajax](#)
[Form Submit Without Refresh Page](#) [Laravel Send Email with Example](#) [Laravel tutorial](#) [Laravel Upload](#)
[Image Using Ajax Tutorial](#) [Mail Send in Laravel with](#)

-8%	-1%		-4%	-51%
-----	-----	--	-----	------