Build Node.js Rest APIs with Express & MySQL

📅 Last modified: August 17, 2021 (https://www.bezkoder.com/node-js-rest-api-express-mysql/)   👤
bezkoder (https://www.bezkoder.com/author/bezkoder/)   📁 Node.js
(https://www.bezkoder.com/category/node-js/)

Express (https://expressjs.com/) is one of the most popular web frameworks for Node.js
that supports routing, middleware, view system… This tutorial will guide you through the
steps of building Node.js Restful CRUD API using Express and interacting with MySQL
database.

Before reading the tutorial, please install MySQL in your machine. The installation
instructions can be found at Official MySQL installation manual
(https://dev.mysql.com/doc/refman/5.7/en/installing.html).

Related Posts:
– Node.js Rest APIs example with Express & MySQL (including Sequelize)
(https://bezkoder.com/node-js-express-sequelize-mysql/)
– Node.js: Upload/Import Excel file data into MySQL Database (https://bezkoder.com/node-
js-upload-excel-file-database/)
– Node.js: Upload CSV file data into MySQL Database (https://bezkoder.com/node-js-
upload-csv-file-database/)

Fullstack:
– Vue.js + Node.js + Express + MySQL example (https://bezkoder.com/vue-js-node-js-
express-mysql-crud-example/)
– Vue.js + Node.js + Express + MongoDB example (https://bezkoder.com/vue-node-
express-mongodb-mevn-crud/)
– Angular 8 + Node.js Express + MySQL example (https://bezkoder.com/angular-node-
express-mysql/)
– Angular 10 + Node.js Express + MySQL example (https://bezkoder.com/angular-10-node-
js-express-mysql/)
– Angular 11 + Node.js Express + MySQL example (https://bezkoder.com/angular-11-node-
js-express-mysql/)
– Angular 12 + Node.js Express + MySQL example (https://bezkoder.com/angular-12-node-
js-express-mysql/)
– React + Node.js + Express + MySQL example (https://bezkoder.com/react-node-express-
mysql/)

Security: Node.js – JWT Authentication & Authorization example
(https://bezkoder.com/node-js-jwt-authentication-mysql/)
Deployment:
– Deploying/Hosting Node.js app on Heroku with MySQL database
(https://bezkoder.com/deploy-node-js-app-heroku-cleardb-mysql/)
– Dockerize Node.js Express and MySQL example – Docker Compose
(https://www.bezkoder.com/docker-compose-nodejs-mysql/)

Node.js & MySQL Associations:
– One-to-Many Relationship example (https://bezkoder.com/sequelize-associate-one-to-many/)
– Many-to-Many Relationship example (https://bezkoder.com/sequelize-associate-many-to-many/)

Contents [hide]

Application overview

We will build Rest Apis for creating, retrieving, updating & deleting Customers.

First, we start with an Express web server. Next, we add configuration for MySQL database,
create Customer model, write the controller. Then we define routes for handling all CRUD
operations:

Methods	Urls	Actions
GET	/customers	get all Customers
GET	/customers/42	get Customer with id=42
POST	/customers	add new Customer
PUT	/customers/42	update Customer with id=42
DELETE	/customers/42	remove Customer with id=42
DELETE	/customers	remove all Customers

Finally, we're gonna test the Rest Apis using Postman.

Our project structure will be like:

```
∨ NODEJS-EXPRESS-MYSQL
  ∨ app
    ∨ config
      JS db.config.js
    ∨ controllers
      JS customer.controller.js
    ∨ models
      JS customer.model.js
      JS db.js
    ∨ routes
      JS customer.routes.js
  > node_modules
  {} package-lock.json
  {} package.json
  JS server.js
```
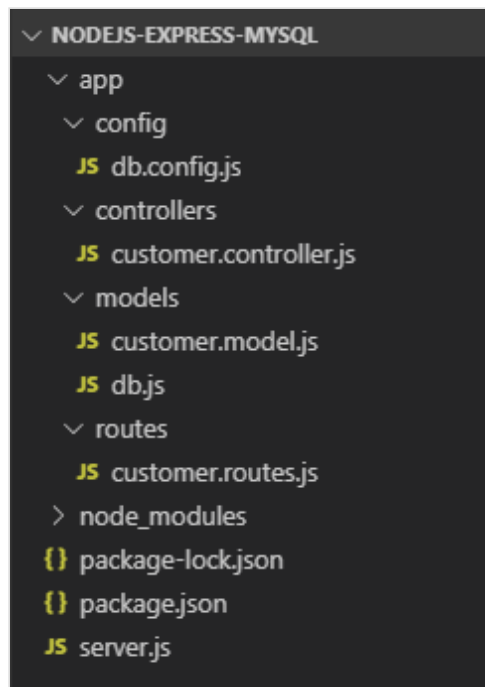
Create Node.js application

Open terminal/console, then create a folder for our application:

```
$ mkdir nodejs-express-mysql
$ cd nodejs-express-mysql
```

Initialize the Node.js application with a *package.json* file: ☒

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

```
npm init

name: (nodejs-express-mysql)
version: (1.0.0)
description: Node.js Restful CRUD API with Node.js, Express and MySQL
entry point: (index.js) server.js
test command:
git repository:
keywords: nodejs, express, mysql, restapi
author: bezkoder
license: (ISC)

Is this ok? (yes) yes
```

Next, we need to install necessary modules: `express`, `mysql` and `body-parser`.
Run the command:

```
npm install express mysql body-parser --save
```

The *package.json* file should look like this:

```
{
  "name": "nodejs-express-mysql",
  "version": "1.0.0",
  "description": "Node.js Restful CRUD API with Node.js, Express and MySQL",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "nodejs",
    "express",
    "mysql",
    "restapi"
  ],
  "author": "bezkoder",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "express": "^4.17.1",
    "mysql": "^2.17.1"
  }
}
```

Setup Express web server

Now, in the root folder, we create a new file named *server.js*:

```
const express = require("express");
const bodyParser = require("body-parser");

const app = express();

// parse requests of content-type: application/json
app.use(bodyParser.json());

// parse requests of content-type: application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({ extended: true }));

// simple route
app.get("/", (req, res) => {
  res.json({ message: "Welcome to bezkoder application." });
});

// set port, listen for requests
app.listen(3000, () => {
  console.log("Server is running on port 3000.");
});
```

What we do are:
– import `express` and `body-parser` modules. Express is for building the Rest apis, and body-parser (https://www.npmjs.com/package/body-parser) helps to parse the request and create the `req.body` object that we will need to access in our routes.
– create an Express app, then add `body-parser` middlewares using `app.use()` method.
– define a GET route which is simple for test.
– listen on port 3000 for incoming requests.

Now we can run the app with command: `node server.js` .
Open your browser, enter the url http://localhost:3000/ (http://localhost:3000/), you will see:
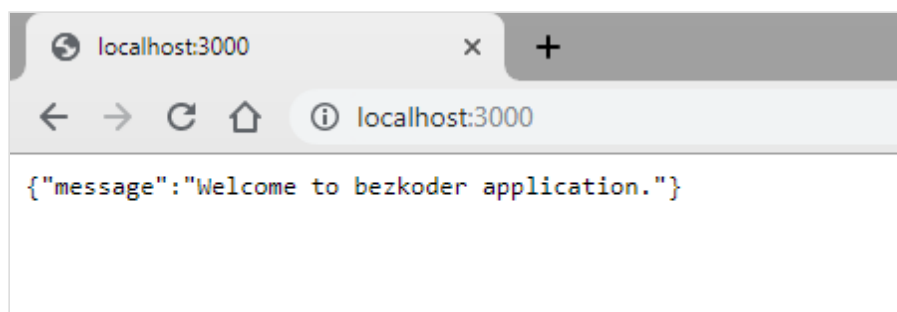
We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

Create MySQL table

Before connecting Node.js Application with MySQL, we need a table first.
So run the SQL script below to create `customers` table:

```
CREATE TABLE IF NOT EXISTS `customers` (
  id int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  email varchar(255) NOT NULL,
  name varchar(255) NOT NULL,
  active BOOLEAN DEFAULT false
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Configure & Connect to MySQL database

We're gonna have a separate folder for configuration. Let's create *config* folder in the *app* folder, under application root folder, then create *db.config.js* file inside that *config* folder with content like this:

```
module.exports = {
  HOST: "localhost",
  USER: "root",
  PASSWORD: "123456",
  DB: "testdb"
};
```

Now create a database connection that uses configuration above.
The file for connection is *db.js*, we put it in *app/models* folder that will contain model in the next step.

X

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

```
const mysql = require("mysql");
const dbConfig = require("../config/db.config.js");

// Create a connection to the database
const connection = mysql.createConnection({
  host: dbConfig.HOST,
  user: dbConfig.USER,
  password: dbConfig.PASSWORD,
  database: dbConfig.DB
});

// open the MySQL connection
connection.connect(error => {
  if (error) throw error;
  console.log("Successfully connected to the database.");
});

module.exports = connection;
```

Define the Model

In *models* folder, create a file called *customer.model.js*. We're gonna define constructor for
`Customer` object here, and use the database connection above to write CRUD functions:

- create a new Customer
- find a Customer by id
- get all Customers
- update a Customer by id
- remove a Customer
- remove all Customers

This is the content inside *customer.model.js*:

```javascript
const sql = require("./db.js");


// constructor
const Customer = function(customer) {
  this.email = customer.email;
  this.name = customer.name;
  this.active = customer.active;
};

Customer.create = (newCustomer, result) => {
  sql.query("INSERT INTO customers SET ?", newCustomer, (err, res) => {
    if (err) {
      console.log("error: ", err);
      result(err, null);
      return;
    }

    console.log("created customer: ", { id: res.insertId, ...newCustomer });
    result(null, { id: res.insertId, ...newCustomer });
  });
};

Customer.findById = (customerId, result) => {
  sql.query(`SELECT * FROM customers WHERE id = ${customerId}`, (err, res) => {
    if (err) {
      console.log("error: ", err);
      result(err, null);
      return;
    }

    if (res.length) {
      console.log("found customer: ", res[0]);
      result(null, res[0]);
      return;
    }

    // not found Customer with the id
    result({ kind: "not_found" }, null);
  });
};

Customer.getAll = result => {
  sql.query("SELECT * FROM customers", (err, res) => {
    if (err) {
      console.log("error: ", err);
      result(null, err);
```

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

X

```javascript
      return;
    }


    console.log("customers: ", res);
    result(null, res);
  });
};


Customer.updateById = (id, customer, result) => {
  sql.query(
    "UPDATE customers SET email = ?, name = ?, active = ? WHERE id = ?",
    [customer.email, customer.name, customer.active, id],
    (err, res) => {
      if (err) {
        console.log("error: ", err);
        result(null, err);
        return;
      }


      if (res.affectedRows == 0) {
        // not found Customer with the id
        result({ kind: "not_found" }, null);
        return;
      }


      console.log("updated customer: ", { id: id, ...customer });
      result(null, { id: id, ...customer });
    }
  );
};


Customer.remove = (id, result) => {
  sql.query("DELETE FROM customers WHERE id = ?", id, (err, res) => {
    if (err) {
      console.log("error: ", err);
      result(null, err);
      return;
    }


    if (res.affectedRows == 0) {
      // not found Customer with the id
      result({ kind: "not_found" }, null);
      return;
```

```javascript
    console.log("deleted customer with id: ", id);
```

```
        result(null, res);
      });
    };

    Customer.removeAll = result => {
      sql.query("DELETE FROM customers", (err, res) => {
        if (err) {
          console.log("error: ", err);
          result(null, err);
          return;
        }

        console.log(`deleted ${res.affectedRows} customers`);
        result(null, res);
      });
    };

    module.exports = Customer;
```

`Customer` model is simple, it contains fields: `email`, `name` & `active`.

We use database connection `query()` method to execute MySQL script: INSERT, SELECT, UPDATE, DELETE. You can find more details about `mysql` module at: https://www.npmjs.com/package/mysql (https://www.npmjs.com/package/mysql).

Define Routes

When a client sends request for an endpoint using HTTP request (GET, POST, PUT, DELETE), we need to determine how the server will reponse. It's why we're gonna setup the routes.

These are routes we define:

- `/customers` : GET, POST, DELETE
- `/customers/:customerId` : GET, PUT, DELETE

Create a *routes* folder inside *app* folder, then create *customer.routes.js* file with content like this:

```
module.exports = app => {
  const customers = require("../controllers/customer.controller.js");

  // Create a new Customer
  app.post("/customers", customers.create);

  // Retrieve all Customers
  app.get("/customers", customers.findAll);

  // Retrieve a single Customer with customerId
  app.get("/customers/:customerId", customers.findOne);

  // Update a Customer with customerId
  app.put("/customers/:customerId", customers.update);

  // Delete a Customer with customerId
  app.delete("/customers/:customerId", customers.delete);

  // Create a new Customer
  app.delete("/customers", customers.deleteAll);
};
```

You can see that we use a controller from `/controllers/customer.controller.js` . It contains methods for handling CRUD operations and will be created in the next step.

We also need to include routes in *server.js* (right before `app.listen()`):

```
...

require("./app/routes/customer.routes.js")(app);

app.listen(...);
```

Create the Controller

Now we create a *controllers* folder inside *app* folder, then we have a file named *customer.controller.js*. Our controller will be written inside this with CRUD functions:

- create
- findAll
- findOne

[X]

We use cookies to improve your experience with the site. To find out more, you can read the full
- update
- delete
  Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept
- deleteAll

```javascript
const Customer = require("../models/customer.model.js");

// Create and Save a new Customer
exports.create = (req, res) => {

};

// Retrieve all Customers from the database.
exports.findAll = (req, res) => {

};

// Find a single Customer with a customerId
exports.findOne = (req, res) => {

};

// Update a Customer identified by the customerId in the request
exports.update = (req, res) => {

};

// Delete a Customer with the specified customerId in the request
exports.delete = (req, res) => {

};

// Delete all Customers from the database.
exports.deleteAll = (req, res) => {

};
```

Let's implement these functions.

Create a new object

Create and Save a new Customer:

```javascript
exports.create = (req, res) => {
  // Validate request
  if (!req.body) {
    res.status(400).send({
      message: "Content can not be empty!"
    });
  }

  // Create a Customer
  const customer = new Customer({
    email: req.body.email,
    name: req.body.name,
    active: req.body.active
  });

  // Save Customer in the database
  Customer.create(customer, (err, data) => {
    if (err)
      res.status(500).send({
        message:
          err.message || "Some error occurred while creating the Customer."
      });
    else res.send(data);
  });
};
```

Retrieve all objects

Retrieve all Customers from the database:

```javascript
exports.findAll = (req, res) => {
  Customer.getAll((err, data) => {
    if (err)
      res.status(500).send({
        message:
          err.message || "Some error occurred while retrieving customers."
      });
    else res.send(data);
  });
};
```

[X]

Retrieve a single object

We use cookies to improve your experience with the site. To find out more, you can read the full

Find a single Customer with a customerId :

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

```javascript
exports.findOne = (req, res) => {
  Customer.findById(req.params.customerId, (err, data) => {
    if (err) {
      if (err.kind === "not_found") {
        res.status(404).send({
          message: `Not found Customer with id ${req.params.customerId}.`
        });
      } else {
        res.status(500).send({
          message: "Error retrieving Customer with id " + req.params.customerId
        });
      }
    } else res.send(data);
  });
};
```

Update an object

Update a Customer identified by the `customerId` in the request:

```javascript
exports.update = (req, res) => {
  // Validate Request
  if (!req.body) {
    res.status(400).send({
      message: "Content can not be empty!"
    });
  }

  Customer.updateById(
    req.params.customerId,
    new Customer(req.body),
    (err, data) => {
      if (err) {
        if (err.kind === "not_found") {
          res.status(404).send({
            message: `Not found Customer with id ${req.params.customerId}.`
          });
        } else {
          res.status(500).send({
            message: "Error updating Customer with id " + req.params.customerId
          });
        }
      } else res.send(data);
    }
  );
};
```

Delete an object

Delete a Customer with the specified `customerId` in the request:

```
exports.delete = (req, res) => {
  Customer.remove(req.params.customerId, (err, data) => {
    if (err) {
      if (err.kind === "not_found") {
        res.status(404).send({
          message: `Not found Customer with id ${req.params.customerId}.`
        });
      } else {
        res.status(500).send({
          message: "Could not delete Customer with id " + req.params.customerId
        });
      }
    } else res.send({ message: `Customer was deleted successfully!` });
  });
};
```

Delete all objects

Delete all Customers from the database:

```
exports.deleteAll = (req, res) => {
  Customer.removeAll((err, data) => {
    if (err)
      res.status(500).send({
        message:
          err.message || "Some error occurred while removing all customers."
      });
    else res.send({ message: `All Customers were deleted successfully!` });
  });
};
```

Test the APIs

Run our Node.js application with command: `node server.js` .
The console shows:

```
Server is running on port 3000.
Successfully connected to the database.
```

Using Postman, we're gonna test all the Apis above.

1. Create a new Customer using `POST /customers` **Api**

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

After creating some new Customers, we can check MySQL table:

```
mysql> SELECT * FROM customers;
+----+--------------------+--------+--------+
| id | email              | name   | active |
+----+--------------------+--------+--------+
|  1 | bezkoder@gmail.com | zKoder |      1 |
|  2 | jack123@gmail.com  | Jack   |      0 |
|  3 | drhelen@gmail.com  | Helen  |      0 |
+----+--------------------+--------+--------+
```

2. **Retrieve all Customers using** `GET /customers` **Api**

```
GET          ▼    http://localhost:3000/customers

Params    Authorization    Headers (7)    Body    Pre-request Script    Tests    Settings

Query Params

  KEY                              VALUE                              DESCRIPTION
  Key                              Value                              Description

Body    Cookies    Headers (6)    Test Results           Status: 200 OK    Time: 39ms

  Pretty    Raw    Preview    Visualize BETA    JSON  ▼    ⇄

   1  [
   2      {
   3          "id": 1,
   4          "email": "bezkoder@gmail.com",
   5          "name": "zKoder",
   6          "active": 1
   7      },
   8      {
   9          "id": 2,
  10          "email": "jack123@gmail.com",
  11          "name": "Jack",
  12          "active": 0
  13      },
  14      {
  15          "id": 3,
  16          "email": "drhelen@gmail.com",
  17          "name": "Helen",
  18          "active": 0
  19      }
  20  ]
```

3. **Retrieve a single Customer by id using** `GET /customers/:customerId` **Api**

```
GET          ▼    http://localhost:3000/customers/2

Params    Authorization    Headers (7)    Body    Pre-request Script    Tests    Settings

Query Params

  KEY                              VALUE                              DESCRIPTION
  Key                              Value                              Description

Body    Cookies    Headers (6)    Test Results           Status: 200 OK    Time: 31ms

  Pretty    Raw    Preview    Visualize BETA    JSON  ▼    ⇄

   1  {
   2      "id": 2,
   3      "email": "jack123@gmail.com",
   4      "name": "Jack",
   5      "active": 0
   6  }
```

4. **Update a Customer using** `PUT /customers/:customerId` **Api**

```
PUT  ▼      http://localhost:3000/customers/3

Params    Authorization    Headers (9)    Body ●    Pre-request Script    Tests    Settings

● none   ● form-data   ● x-www-form-urlencoded   ● raw   ● binary   ● GraphQL BETA   JSON  ▼

1 ▼ {
2       "email": "drhelen@gmail.com",
3       "name": "Dr.Helen",
4       "active": true
5   }
```

```
Body   Cookies   Headers (6)   Test Results          Status: 200 OK   Time: 306ms

Pretty   Raw   Preview   Visualize BETA    JSON  ▼

1   {
2       "id": "3",
3       "email": "drhelen@gmail.com",
4       "name": "Dr.Helen",
5       "active": true
6   }
```

Check `customers` table after a row was updated:

```
mysql> SELECT * FROM customers;
+----+--------------------+----------+--------+
| id | email              | name     | active |
+----+--------------------+----------+--------+
|  1 | bezkoder@gmail.com | zKoder   |      1 |
|  2 | jack123@gmail.com  | Jack     |      0 |
|  3 | drhelen@gmail.com  | Dr.Helen |      1 |
+----+--------------------+----------+--------+
```

5. **Delete a Customer using** `DELETE /customers/:customerId` **Api**

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

Customer with id=2 was removed from `customers` table:

```
mysql> SELECT * FROM customers;
+----+--------------------+----------+--------+
| id | email              | name     | active |
+----+--------------------+----------+--------+
|  1 | bezkoder@gmail.com | zKoder   |      1 |
|  3 | drhelen@gmail.com  | Dr.Helen |      1 |
+----+--------------------+----------+--------+
```

6. **Delete all Customers using** `DELETE /customers` **Api**

We use cookies to improve your experience with the site. To find out more, you can read the full

Now there are no rows in customers table:

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

```
mysql> SELECT * FROM customers;
Empty set (0.00 sec)
```

Conclusion

Today, we've learned how to create Node.js Rest Apis with an Express web server. We also know way to add configuration for MySQL database, create a model, write a controller and define routes for handling all CRUD operations.

You can find more interesting thing in the next tutorial:
– Deploying/Hosting Node.js app on Heroku with MySQL database
(https://bezkoder.com/deploy-node-js-app-heroku-cleardb-mysql/)
– Dockerize Node.js Express and MySQL example – Docker Compose
(https://www.bezkoder.com/docker-compose-nodejs-mysql/)

If you want to use Sequelize to reduce boilerplate code, there is a post for this:
Node.js Rest APIs example with Express, Sequelize & MySQL (https://bezkoder.com/node-js-express-sequelize-mysql/)

Happy learning! See you again.

Further Reading

- Express.js Routing (https://expressjs.com/en/guide/routing.html)
- https://www.npmjs.com/package/express (https://www.npmjs.com/package/express)
- https://www.npmjs.com/package/body-parser (https://www.npmjs.com/package/body-parser)
- https://www.npmjs.com/package/mysql (https://www.npmjs.com/package/mysql)

– Upload/store images in MySQL using Node.js, Express & Multer
(https://www.bezkoder.com/node-js-upload-image-mysql/)
– Node.js Express File Upload Rest API example using Multer
(https://www.bezkoder.com/node-js-express-file-upload/)

Source code

You can find the complete source code for this example on Github
(https://github.com/bezkoder/nodejs-express-mysql).

crud (https://www.bezkoder.com/tag/crud/)   express (https://www.bezkoder.com/tag/express/)

mysql (https://www.bezkoder.com/tag/mysql/)   node.js (https://www.bezkoder.com/tag/node-js/)   ☒

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

167 thoughts to "Build Node.js Rest APIs with Express & MySQL"

prowebkarawang
October 4, 2019 at 10:36 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-7)

The tutorial is concise and clear and very helpful, thanks.

REPLY

bezkoder
October 6, 2019 at 6:38 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-8)

Hi, thank you 😊
Hope it help you.

REPLY

Suju
September 9, 2020 at 10:48 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4660)

Very helpful, thanks bro.

REPLY

Lesego
September 18, 2020 at 8:55 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4774)

Thank you very much, this was very helpful. From now on my project will be structured in this fashion

REPLY

X

We use cookies to improve your experience with the site. To find out more, you can read the full
Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

Roger
January 26, 2021 at 9:42 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-6566)

Hi. This is good stuff, what's the name of this pattern?

REPLY

KIRANDEEP KAUR
August 4, 2020 at 12:24 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4199)

where to create the object as I am learning don't have much knowledge

REPLY

Federico
October 24, 2019 at 2:21 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-25)

You're really awesome bro!

REPLY

bezkoder
October 24, 2019 at 3:18 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-26)

Hi, thank you for your comment! ?
Hope this help.

REPLY

gus abas
October 27, 2019 at 12:24 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-31)

i have a problem when i try, the message is "cannot find module
'..controllers/customer.contorller.js"
please help

REPLY

X

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept
bezkoder

October 27, 2019 at 1:43 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-32)

Hi, there is a typo: `contorller` . It should be `customer.controller.js` .

REPLY

---

fobb
April 9, 2020 at 3:11 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-806)

I still get error of reading req.body

REPLY

---

bezkoder
April 10, 2020 at 12:16 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-812)

Make sure that you set `'Content-Type': 'application/json'` for HTTP request Header.

REPLY

Pankaj
April 22, 2020 at 10:55 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1207)

Cannot POST /customers when testing api

---

bezkoder
April 22, 2020 at 10:56 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1285)

Maybe you forgot to add Content Type to HTTP header.

Junior
December 16, 2020 at 4:11 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5816)

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

I was wrong to declare the body-parser after the routes. Try this:

[X]

```
const app = express();
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
require("./app/routes/customer.routes.js")(app);
```

REPLY

Roger

October 29, 2019 at 9:05 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-33)

I got a problem when i post a request to save new customers

```
_http_outgoing.js:470
    throw new ERR_HTTP_HEADERS_SENT('set');
    ^

Error [ERR_HTTP_HEADERS_SENT]: Cannot set headers after they are sent to t
    at ServerResponse.setHeader (_http_outgoing.js:470:11)
    at ServerResponse.header (D:\NodeJS\webtuthien\node_modules\express\li
    at ServerResponse.send (D:\NodeJS\webtuthien\node_modules\express\lib\
    at done (D:\NodeJS\webtuthien\node_modules\express\lib\response.js:100
    at Immediate. (D:\NodeJS\webtuthien\node_modules\express-handlebars\li
    at runCallback (timers.js:705:18)
    at tryOnImmediate (timers.js:676:5)
    at processImmediate (timers.js:658:5)
```

REPLY

bezkoder

October 29, 2019 at 2:47 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-35)

Hi, please show me more details about your HTTP POST request.

REPLY

[X]

We use cookies to improve your experience with the site. To find out more, you can read the full

Bhargavi Balusu

Privacy & Policy (https://bezkoder.com/privacy-policy/) ss-mysql/#comment-3099)

Accept

Hi,

What are the headers you have used. because i tried to do POST, but got this message,

{

"message": "ER_BAD_NULL_ERROR: Column 'username' cannot be null"

}

REPLY

bezkoder

June 4, 2020 at 1:22 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3103)

Hi, it is `Content-type: application/json` 🙂

REPLY

golavo

November 7, 2019 at 9:46 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-37)

Very helpful, thanks for this tutorial.

REPLY

Axel

December 31, 2019 at 4:51 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-128)

Hello

im dealing with another issue and can't figure it out

Server is running on port 3000.

Successfully connected to the database.

error: Error: ER_BAD_NULL_ERROR: Column 'email' cannot be null

at Query.Sequence._packetToError (/home/axel/nodejs-express-

mysql/node_modules/mysql/lib/protocol/sequences/Sequence.js:47:14)

at Query.ErrorPacket (/home/axel/nodejs-express-

mysql/node_modules/mysql/lib/protocol/sequences/Query.js:77:18)

at Protocol._parsePacket (/home/axel/nodejs-express-

mysql/node_modules/mysql/lib/protocol/Protocol.js:291:23)

at Parser._parsePacket (/home/axel/nodejs-express-

```
mysql/node_modules/mysql/lib/protocol/Parser.js:433:10)
  at Parser.write (/home/axel/nodejs-express-
mysql/node_modules/mysql/lib/protocol/Parser.js:43:10)
  at Protocol.write (/home/axel/nodejs-express-
mysql/node_modules/mysql/lib/protocol/Protocol.js:38:16)
  at Socket. (/home/axel/nodejs-express-
mysql/node_modules/mysql/lib/Connection.js:91:28)
  at Socket. (/home/axel/nodejs-express-
mysql/node_modules/mysql/lib/Connection.js:525:10)
  at Socket.emit (events.js:305:20)
  at addChunk (_stream_readable.js:341:12)
  —————————–
  at Protocol._enqueue (/home/axel/nodejs-express-
mysql/node_modules/mysql/lib/protocol/Protocol.js:144:48)
  at Connection.query (/home/axel/nodejs-express-
mysql/node_modules/mysql/lib/Connection.js:201:25)
  at Function.Customer.create (/home/axel/nodejs-express-
mysql/app/models/customer.model.js:11:7)
  at exports.create (/home/axel/nodejs-express-
mysql/app/controllers/customer.controller.js:20:12)
  at Layer.handle [as handle_request] (/home/axel/nodejs-express-
mysql/node_modules/express/lib/router/layer.js:95:5)
  at next (/home/axel/nodejs-express-
mysql/node_modules/express/lib/router/route.js:137:13)
  at Route.dispatch (/home/axel/nodejs-express-
mysql/node_modules/express/lib/router/route.js:112:3)
  at Layer.handle [as handle_request] (/home/axel/nodejs-express-
mysql/node_modules/express/lib/router/layer.js:95:5)
  at /home/axel/nodejs-express-mysql/node_modules/express/lib/router/index.js:281:22
  at Function.process_params (/home/axel/nodejs-express-
mysql/node_modules/express/lib/router/index.js:335:12) {
  code: 'ER_BAD_NULL_ERROR',
  errno: 1048,
  sqlMessage: "Column 'email' cannot be null",
  sqlState: '23000',
  index: 0,
  sql: 'INSERT INTO customers SET `email` = NULL, `name` = NULL, `active` = NULL'
}
```

Thanks

REPLY

bezkoder

January 1, 2020 at 1:12 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-130)

Hi, please tell me which HTTP request along with payload you sent to the server.

REPLY

Rosapink

January 25, 2020 at 6:30 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-254)

You have to set headers for content-type: application/json & content-type: x-www-form-urlencoded

REPLY

Deni

January 30, 2020 at 1:17 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-272)

my problem same with Axel, how to fix it? please…

REPLY

bezkoder

January 30, 2020 at 8:12 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-273)

Maybe you forgot to set headers for content-type: application/json & content-type: x-www-form-urlencoded on REST Client requests 🙂

REPLY

hope

February 5, 2020 at 6:55 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-304)

Where do I set headers? I'm confused

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)　　　Accept

bezkoder
February 6, 2020 at 2:20 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-305)

Hi, for example, with Postman, you can see **Header** tab (besides **Body** tab) where we set header for HTTP request 🙂

Yasir
April 27, 2020 at 10:22 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1680)

Thanks r million

REPLY

Roberta
January 13, 2020 at 8:42 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-186)

Thank You for the tutorial! Helped a lot!

REPLY

Roberta
January 14, 2020 at 12:59 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-194)

For making many to many relationship need to use Sequelize or can make it without it?

REPLY

bezkoder
January 16, 2020 at 7:58 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-206)

Yes, you can make many-to-many relationship with table structure as Sequelize. But you will need to write more boilerplate.

REPLY

X

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

Nick
February 6, 2020 at 9:53 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-306)

Thank you for the tutorial, it really helps me a lot! I'm still a beginner so there are some parts I don't understand.
Can you please explain to me what the "result" function does in the models? And why does it need two parameters?

REPLY

bezkoder
February 6, 2020 at 10:06 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-307)

Hi, one for data, and one for error data 🙂

REPLY

Nick
February 6, 2020 at 10:12 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-308)

Where do the data go from there? Are they linked to the controller?
Thanks in advance!

REPLY

bezkoder
February 7, 2020 at 3:48 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-327)

Yeah, they are linked to the controller.

For example:

```
Customer.getAll = result => {
  ...
  result(null, err);
  ...
};
```

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

Now look at functions in controller:

```
Customer.getAll((err, data) => { ... })
```

You can see that we use `(err, data) => { ... }` as `result`.

The `result` parameter of `getAll()` is a function.

In other words, we pass a function to `getAll()` as parameter.

REPLY

Nick

February 10, 2020 at 7:27 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-334)

Thanks for the explanation! 🙂

bezkoder

February 10, 2020 at 10:03 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-336)

You're welcome 🙂

Agustin Martinez

July 1, 2021 at 12:33 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-9381)

Shouldn't it be, in case of error, like this: result (err, null). ?

NewBie

May 25, 2020 at 6:55 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2931)

coming from OOP where all variables are strictly defined for certain type, for example `DbConnection x`, it's very hard to adapt to Node's free-style arbitrary declaration and what it implies. Do you have any articles to help jump-start?

Thank you very much in advanced.

We use cookies to improve your experience with the site. To find out more, you can read the full

REPLY

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

bezkoder

May 26, 2020 at 4:11 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2937)

Hi, thanks for your suggestion.

Currently I don't have much time to write many tutorials that you and other people need (although I really want to do).

I will try to work for new Typescript style in the future ☺

REPLY

Santhosh sandy

October 25, 2020 at 6:37 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5192)

Firstly Thank you for all your beautiful tutorials. If you got time, Please try to post a Node.js, MySQL RestApi CRUD app using typescript.

Jo

February 9, 2020 at 6:57 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-332)

Thanks Bezkoder – great tutorial, super helpful!

REPLY

sami

February 17, 2020 at 10:03 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-367)

Great tutorial. Can we add user authentication layer with jwt to this customers module?

REPLY

bezkoder

February 17, 2020 at 11:14 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-370)

Hi, you can find how to implement authentication at the post:
Node.js & JWT – Token Based Authentication & Authorization example
(https://bezkoder.com/node-js-jwt-authentication-mysql/)

[X]

abn
February 24, 2020 at 9:40 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-406)

The tutorial is concise and clear and very helpful, thanks.

yet If there another table 'Product' so that a customer has one or many product
and I want to return a customer with its products, how could I proceed?

bezkoder
February 24, 2020 at 10:45 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-407)

Hi, I will write a tutorial for your suggestion when having time 😊

CoolKat
March 15, 2020 at 2:12 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-537)

I cloned the project from GitHub, and created the MySQL table exactly like it was done in
the tutorial, but when I try to post a new customer using Postman, I receive the error:

{
"message": "ER_BAD_NULL_ERROR: Column 'email' cannot be null"
}

If I manually put data into MySQL and perform a GET request, I get the data correct data,
so something seems to not be working with POST.

bezkoder
March 17, 2020 at 2:24 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-551)

Hi, did you set appropriate Header: Content-Type: application/json when sending
POST request?

REPLY

anjana
June 30, 2020 at 10:45 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3539)

Did you find solution?

REPLY

joseph
June 13, 2021 at 6:45 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-9061)

in postman, next to the "graphQL", by default it says "text". make clic that "Text" optioon and shoose "Json". I had the same mistake and it fixed it. note that in the screenshot it already says "JSON" so you have to make it look like that on your postman

REPLY

Vadim
March 17, 2020 at 9:36 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-552)

Great tutorial! Exactly what I've been searching for. Thank you!

REPLY

Ebadullah
March 23, 2020 at 6:08 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-609)

i got a problem when i get a customer by id
"message": " Error retrieving customer with id 2"
please help

REPLY

X

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

bezkoder

March 24, 2020 at 3:00 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-616)

Hi, please make sure that there is a Customer with id=2.

REPLY

minato

March 26, 2020 at 6:52 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-653)

Thank you for this Node CRUD with MySQL tutorial, very inspiring!

REPLY

Febrian

March 29, 2020 at 8:18 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-672)

Great tutorial and very helpful. thank you ☺

REPLY

Kayode

March 31, 2020 at 1:53 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-695)

Very straight forward and helpful. Thank you.

REPLY

Iron Man

April 4, 2020 at 4:45 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-725)

Great tutorial. I am not even Node developer but i got the Job done. Thanks A lots man

REPLY

ⓧ

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

Rohit Nikam (Rowhit)

April 8, 2020 at 6:05 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-781)

This is very nice and clean and neat tutorial, thank you!

but i have one error, After run node server.js command it shows:

Server is running on port no 3000

Successfully connected to the database!

Using Postman, i test the create new Customers APIs it gives me error in response:

{

"message": "ER_NO_DB_ERROR: No database selected"

}

Please kindly share your response.

REPLY

Rohit Nikam (Rowhit)

April 8, 2020 at 6:52 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-783)

In node mysql, the properties are called database, not db.

// Create a connection to the database

```
const connection = mysql.createConnection({
  host: dbConfig.HOST,
  user: dbConfig.USER,
  password: dbConfig.PASSWORD,
  db: dbConfig.DB
});
```

So, change your config to:

```
// Create a connection to the database
const connection = mysql.createConnection({
  host: dbConfig.HOST,
  user: dbConfig.USER,
  password: dbConfig.PASSWORD,
  database: dbConfig.DB    //This was changd
});
```

X

We use cookies to improve your experience with the site. To find out more, you can read the full
So sorry it's my mistake, and thanks again for this tutorial.

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

REPLY

bezkoder

April 8, 2020 at 10:24 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-788)

Yeah, I'm so happy to hear that 🙂

REPLY

Rohit Nikam (Rowhit)

April 9, 2020 at 11:45 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-801)

Hey bezkoder,
You have any tutorial for promise and promise chain concept using node

REPLY

bezkoder

April 11, 2020 at 2:44 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-828)

Hi, I will write them when having time 🙂

Therese

April 9, 2020 at 11:49 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-802)

Hi! Thanks for a very useful tutorial. Im currently creating a React.js project and i am going to use this tutorial connecting to the client side. Right now i have one frontend folder and this backend folder (this tutorial) but are unsure the best way to connect those two? Thankful for any help..

REPLY

bezkoder

April 10, 2020 at 1:35 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-813)

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

[X]

Hi, you can find the fullstack example that uses React as frontend aand Node.js Express as backend at the post:

React + Node.js + Express + MySQL example: Build a CRUD App (https://bezkoder.com/react-node-express-mysql/)

REPLY

Bruder

April 22, 2020 at 9:57 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1280)

Very clear, easy to implement step by step. This is one of the best Node.js with MySQL tutorials I've ever seen.
Thank you!

REPLY

Emil Mammadov

April 23, 2020 at 4:25 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1413)

Thank you very much. That was the best.

REPLY

Nixon Augustine

April 29, 2020 at 11:13 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1763)

Hi, Wonderful tutorial. Really Appreciate the time you took to write the Tutorial.
I have doubt

require("./app/routes/customer.routes.js")(app);

is this statement correct ? why is require not assigned to a variable. And (app) is mentioend at the end ?

REPLY

Hi, you can see that in *app/routes/customer.routes.js*, we export a function:

```
module.exports = app => {
  ...
}

// or:
const func = app => {
  ...
}

module.exports = func;
```

So the `require("./app/routes/customer.routes.js")(app)` is equivalent to:

```
const routeFunction = require("./app/routes/customer.routes.js");
routeFunction(app);
```

REPLY

NewBie

May 23, 2020 at 11:40 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2837)

I'm a newbie, still don't get it. routeFunction() takes parameter?
my understanding is require() is the combination of declaration and instantiation of a class, for example
myObj app = myObj();

REPLY

---

bezkoder

May 23, 2020 at 11:49 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2839)

A function is an object, so it can be a function 🙂

REPLY

X

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

NewBie
May 25, 2020 at 5:33 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2928)

Thank you for explaining. So the two `app` are different.
can `module.exports = app => {` be interpreted as `module.exports = (app) => {`?

bezkoder
May 26, 2020 at 4:09 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2936)

Yeah, they are the same 🙂

NewBie
May 23, 2020 at 11:55 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2841)

or, is it equivalent to this?
const routeFunction = require("./app/routes/customer.routes.js");
app.use('/', routeFunction);

REPLY

bezkoder
May 24, 2020 at 1:37 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2867)

Ah no, you can see that, in `./app/routes/customer.routes.js` we export the function:

```
module.exports = app => {
  app.post("/customers", customers.create);
  app.get("/customers", customers.findAll);
  app.get("/customers/:customerId", customers.findOne);
  app.put("/customers/:customerId", customers.update);
  app.delete("/customers/:customerId", customers.delete);
  app.delete("/customers", customers.deleteAll);
};
```

So, the way we use it is equivalent to:

X

```
const routeFunction = require("./app/routes/customer.routes.js");
routeFunction(app);
```

Then the result will be:

```
app.post("/customers", customers.create);
app.get("/customers", customers.findAll);
app.get("/customers/:customerId", customers.findOne);
app.put("/customers/:customerId", customers.update);
app.delete("/customers/:customerId", customers.delete);
app.delete("/customers", customers.deleteAll);
```

Ali

February 3, 2021 at 3:16 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-6780)

i am also doubtful at this line, and one more thing is

if

require("./app/routes/customer.routes.js")(app);

require("./app/routes/order.routes.js")(app);

data from order.routes.js is not displaying.

why ? and how to declare multiple routes?

REPLY

Nixon Augustine

April 29, 2020 at 11:39 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1764)

This is an error I get. Instead of Customer ,I have user, but the other code is the same. Where have i made the mistake.

TypeError: User.viewAll is not a function   at exports.viewAll (E:\touchtraks\api\v1\controller\user.controller.js:4:7)   at Layer.handle [as handle_request] (E:\touchtraks\api\node_modules\express\lib\router\layer.js:95:5)   at next (E:\touchtraks\api\node_modules\express\lib\router\route.js:137:13)   at Route.dispatch (E:\touchtraks\api\node_modules\express\lib\router\route.js:112:3)   at

We use cookies to improve your experience with the site. To find out more, you can read the full Layer.handle [as handle_request] (E:\touchtraks\api\node_modules\express\lib\router\layer.js:95:5)   at **Privacy & Policy (https://bezkoder.com/privacy-policy/).**   Accept E:\touchtraks\api\node_modules\express\lib\router\index.js:281:22   at

X

Function.process_params
(E:\touchtraks\api\node_modules\express\lib\router\index.js:335:12)   at next
(E:\touchtraks\api\node_modules\express\lib\router\index.js:275:10)   at
urlencodedParser (E:\touchtraks\api\node_modules\body-
parser\lib\types\urlencoded.js:82:7)   at Layer.handle [as handle_request]
(E:\touchtraks\api\node_modules\express\lib\router\layer.js:95:5)

REPLY

bezkoder
April 29, 2020 at 11:42 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1798)

Hi, please show your code here 🙂

REPLY

Nixon Augustine
April 30, 2020 at 5:01 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1805)

Hi, Really Appreciate you answering my previous question. I have a question, which might soud too basic.

It would be great if you could answer it .

Can you explain the flow of control from the time an api request hits the server.

From Server.js -> where does the control go and what does it check.

Thanks in advance.

REPLY

bezkoder
April 30, 2020 at 12:01 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1837)

You can follow this flow:

server.js -> routes -> controller -> model.

REPLY

🅇

Nixon Augustine
April 30, 2020 at 5:35 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1807)

I got the error fixed.

REPLY

Yunus Emre
August 22, 2020 at 10:56 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4434)

Hey Nixon. Just a quick reminder. When you fix the problem, always share the steps that you have take in. You never know how it might help to someone in future.

For anyone who is having the same problem, just add "module.exports.Course = Course;" at the end of course.model.js file. We need to export objects, classes to be able reference and use in in other files.

REPLY

Yunus Emre
August 22, 2020 at 11:03 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4435)

Sorry just add "module.exports = Course"

ahmed hlaimi
May 2, 2020 at 12:01 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1862)

hello awsome tuto bu u have one mistake in require("./app/routes/customer.routes.js")
(app);
u should delete app from path like this :
require("./routes/customer.routes.js")(app); at server.js

REPLY

We use cookies to improve your experience with the site. To find out more, you can read the full

bezkoder
Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept
May 2, 2020 at 12:22 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1864)

[X]

Hi, because we put *server.js* in the root folder, so the directory should be
"./app/routes/customer.routes.js". 😊

REPLY

sarah
November 2, 2020 at 7:14 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5268)

Hi, i removed /app and it worked for me

REPLY

Karl
May 3, 2020 at 2:10 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1896)

Hi bezkoder,

thks for this tutorial. I m also new to react and try to use this tutorial to train myself but I
cant move forward. Can you tell me where and how I should run the mysql query to
create the table CUSTOMER?

"So run the SQL script below to create customers table:"

I've done it in the root directory but got some errors
Thks in advance

REPLY

bezkoder
May 4, 2020 at 2:14 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-1904)

Hi, you can use MySQL Workbench (https://www.mysql.com/products/workbench/) or
MySQL Client (https://dev.mysql.com/doc/refman/5.7/en/mysql-commands.html) to run
MySQL command.

REPLY

ⓧ

We use cookies to improve your experience with the site. To find out more, you can read the full
Harsh
Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept
May 6, 2020 at 6:18 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2035)

Hey there is a mistake in the code .

// Instead of this

Customer.findById = (customerId, result) => {
sql.query(`SELECT * FROM customers WHERE id = ${customerId}`, (err, res) => {
…..}

// Please correct it to

sql.query(`SELECT * FROM customers WHERE id = "${customerId}"`, (err, res) => {
…..}
// add the double quotes in where clause

Thanks for the tutorial though
Cheers!

REPLY

Bella
May 11, 2020 at 6:19 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2221)

Wow, this is really nice. This is my first workable project with js and it's so impressive. The
next step, how can I link it with my UI? Do you have any nice lesson to suggest? Cheers

REPLY

Sam
May 14, 2020 at 6:19 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2438)

https://bezkoder.com/react-node-express-mysql/ (https://bezkoder.com/react-node-
express-mysql/)
Here you go

REPLY

kkwaxy
May 17, 2020 at 7:28 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2537)

[X]

We use cookies to improve your experience with the site. To find out more, you can read the full
Thanks a lot. It's been very helpful for me. I'm newer on nodejs

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept
REPLY

david arias
May 19, 2020 at 5:06 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2613)

Buenas noches Koder:

Me podrías apoyar con este error… Ya probé varias opciones e investigué otras opciones
y nada.

O si puedes dar otra opción. Por favor.

——
————————-
require("./app/routes/customer.routes")(app)
^

TypeError: require(…) is not a function
at Object. (C:\Cursos\KODER\node-express-mysql\server.js:19:40)
at Module._compile (internal/modules/cjs/loader.js:1185:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:1205:10)
at Module.load (internal/modules/cjs/loader.js:1034:32)
at Function.Module._load (internal/modules/cjs/loader.js:923:14)
at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:71:12)
at internal/main/run_main_module.js:17:47
PS C:\Cursos\KODER\node-express-mysql>

REPLY

bezkoder
May 19, 2020 at 9:46 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2625)

Hi, please show me your code in `./app/routes/customer.routes` .

REPLY

Msukma
May 20, 2020 at 5:27 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2672)

Hai there, thank for this great tutorial!

X

I have issue, on showing html when using view template engine (ejs), on this scenario
1.parsing url with params or more than 1 level url name, but but work well when showing
1 level url

——————————————————————-

app.get('/home:params',function(req,res){

res.render('home')

})

or..

app.get('/home/sub',function(req,res){

res.render('home')

})

thanks,

REPLY

hia

May 23, 2020 at 11:45 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-2838)

I will recommend to all my friends and colleagues. Thank you very much for this tutorial!

REPLY

Michele Bridi

June 5, 2020 at 8:01 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3111)

hello, sorry for the ignorance but I would like to know how to create the db, if I have not
misunderstood you explain how to connect and do the rest but I do not know how to
access / create the db thanks

REPLY

Percy

June 5, 2020 at 10:37 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3113)

Im getting this error"No database selected"

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

X

error: Error: ER_NO_DB_ERROR: No database selected

at Query.Sequence._packetToError
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\protocol\sequence
s\Sequence.js:47:14)

at Query.ErrorPacket
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\protocol\sequence
s\Query.js:79:18)

at Protocol._parsePacket
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\protocol\Protocol.j
s:291:23)

at Parser._parsePacket
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\protocol\Parser.js:
433:10)

at Parser.write
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\protocol\Parser.js:
43:10)

at Protocol.write
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\protocol\Protocol.j
s:38:16)

at Socket.
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\Connection.js:88:2
8)

at Socket.
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\Connection.js:526:
10)

at Socket.emit (events.js:310:20)

at addChunk (_stream_readable.js:286:12)

——————————

at Protocol._enqueue
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\protocol\Protocol.j
s:144:48)

at Connection.query
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\mysql\lib\Connection.js:198:
25)

at Function.User.getAll
(C:\Users\master\BackEnd\DeclareApp\code\app\models\user.model.js:47:7)

at exports.findAll
(C:\Users\master\BackEnd\DeclareApp\code\app\controllers\user.controller.js:34:10)

at Layer.handle [as handle_request]
(C:\Users\master\BackEnd\DeclareApp\code\node_modules\express\lib\router\layer.js:9
5:5)

at next

(C:\Users\master\BackEnd\DeclareApp\code\node_modules\express\lib\router\route.js:1
37:13)

at Route.dispatch

(C:\Users\master\BackEnd\DeclareApp\code\node_modules\express\lib\router\route.js:1
12:3)

at Layer.handle [as handle_request]

(C:\Users\master\BackEnd\DeclareApp\code\node_modules\express\lib\router\layer.js:9
5:5)

at

C:\Users\master\BackEnd\DeclareApp\code\node_modules\express\lib\router\index.js:2
81:22

at Function.process_params

(C:\Users\master\BackEnd\DeclareApp\code\node_modules\express\lib\router\index.js:3
35:12) {

code: 'ER_NO_DB_ERROR',

errno: 1046,

sqlMessage: 'No database selected',

sqlState: '3D000',

index: 0,

sql: 'SELECT * FROM user'

}

REPLY

---

bezkoder

June 7, 2020 at 1:20 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3132)

Hi, please check the database configuration in the code (*db.config.js*). It should match your PC MySQL database configuration.

REPLY

Edward

June 12, 2020 at 9:35 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3243)

Hi, Great Tutorial.

When testing the APIs with postman, I try and create a new customer. It successfully

ReferenceError: result is not defined
at Query. (****/customer.model.js:19:9)

```
This is the code im using to create the customer in the database:
Customer.create = (newCustomer, results) => {
    sql.query("INSERT INTO customers SET ?", newCustomer, (err, res) => {
        if (err) {
          console.log("error: ", err);
          result(err, null);
          return;
        }

        console.log("created customer: ", { id: res.insertId, ...newCustom
        result(null, { id: res.insertId, ...newCustomer });
    });
}
```

◀ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ ▶

Any help would be greatly appreciated

REPLY

324234
June 27, 2020 at 5:29 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3488)

hhmm..may be it should be result, not results here
Customer.create = (newCustomer, result)

REPLY

NewBie
July 18, 2020 at 5:50 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3817)

you have `results` and `result`.

REPLY

[X]

We use cookies to improve your experience with the site. To find out more, you can read the full
Naveen Gupta
Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept
June 13, 2020 at 1:24 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3254)

Thanks, explained so well.

REPLY

yos
June 23, 2020 at 9:43 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3423)

hello, i need to know, if i want to get some customers, not all.

thanks a lot

REPLY

bezkoder
June 24, 2020 at 12:35 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3425)

Hi, if you want to make pagination, please read this tutorial:
Server side Pagination in Node.js with Sequelize & MySQL (https://bezkoder.com/node-js-sequelize-pagination-mysql/)

REPLY

yosmar gonzalez
July 11, 2020 at 10:34 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3701)

Hello bezkoder, i am very new in api rest. i need get some customer for name and not for id.

this is for id in typescript

🗙

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/) Accept

```
public async getOne (req:Request,res:Response): Promise{
        const { id } = req.params;
        const habitante = await pool.query('SELECT * FROM habitantes WH
        if (habitante.length>0)
        {
            return res.json(habitante[0]);
        }
        console.log(habitante);
        res.status(404).json({ text:'El habitante '+ req.params.id + '
    }
```

and the routes is

```
  this.router.get('/:id', HabitantesController.getOne);
```

this works perfectly

i need something like that

```
public async getSome (req:Request,res:Response): Promise{
        const { termino } = req.params;
        const habitantes = await pool.query('SELECT * FROM habitantes W
        if (habitantes.length>0)
        {
            return res.json(habitantes);
        }
                console.log(habitantes);
        res.status(404).json({ text:'No exiten habitantes que coincidan
    }
```

habitantesroutes.ts

```
  this.router.get('/:termino', HabitantesController.getSome);
```

this doesn't work but always the Api consider the parameter 'termino' as if it were the id

help me please.

I do not know if I explain very well, my English is very basic.

regards

REPLY

mamta

June 27, 2020 at 1:49 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3482)

Hello sir, here is my code of server.js
const express = require("express");
const bodyParser = require("body-parser");

const app = express();

// parse requests of content-type: application/json
app.use(bodyParser.json());

// parse requests of content-type: application/x-www-form-urlencoded
app.use(bodyParser.urlencoded({ extended: true }));

// simple route
app.get("/", (req, res) => {
res.json({ message: "Welcome to bezkoder application." });
});
require("./app/routes/customer.routes.js")(app);
// set port, listen for requests
app.listen(3000, () => {
console.log("Server is running on port 3000.");
});

created customer: { id: 3, email: undefined, name: undefined }
where is the problem

REPLY

Vanille

March 16, 2021 at 8:45 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-7626)

Hey, i've got the same problem, did you find a solution?

REPLY

X

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept
gndv

August 17, 2021 at 12:44 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-10942)

Double check that you're sending the request with the proper JSON attached, i.e:

{

"email": "test@email.com",

"name": "Bob".

"active": true

}

REPLY

---

maky

June 28, 2020 at 2:38 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3495)

Thank you very much for this tutorial!

REPLY

---

nitu

July 11, 2020 at 5:25 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3696)

Server is running on port 3000.

D:\nodejs-express-mysql\node_modules\mysql\lib\protocol\Parser.js:437

throw err; // Rethrow non-MySQL errors

^

Error: ER_NOT_SUPPORTED_AUTH_MODE: Client does not support authentication

protocol requested by server; consider upgrading MySQL client

at Handshake.Sequence._packetToError (D:\nodejs-express-

mysql\node_modules\mysql\lib\protocol\sequences\Sequence.js:47:14)

at Handshake.ErrorPacket (D:\nodejs-express-

mysql\node_modules\mysql\lib\protocol\sequences\Handshake.js:123:18)

at Protocol._parsePacket (D:\nodejs-express-

mysql\node_modules\mysql\lib\protocol\Protocol.js:291:23)

at Parser._parsePacket (D:\nodejs-express-

mysql\node_modules\mysql\lib\protocol\Parser.js:433:10)

at Parser.write (D:\nodejs-express-

mysql\node_modules\mysql\lib\protocol\Parser.js:43:10)

at Protocol.write (D:\nodejs-express-

mysql\node_modules\mysql\lib\protocol\Protocol.js:38:16)

at Socket. (D:\nodejs-express-mysql\node_modules\mysql\lib\Connection.js:88:28)

at Socket. (D:\nodejs-express-mysql\node_modules\mysql\lib\Connection.js:526:10)

at Socket.emit (events.js:198:13)

at addChunk (_stream_readable.js:288:12)

——————–

at Protocol._enqueue (D:\nodejs-express-
mysql\node_modules\mysql\lib\protocol\Protocol.js:144:48)

at Protocol.handshake (D:\nodejs-express-
mysql\node_modules\mysql\lib\protocol\Protocol.js:51:23)

at Connection.connect (D:\nodejs-express-
mysql\node_modules\mysql\lib\Connection.js:116:18)

at Object. (D:\nodejs-express-mysql\app\models\db.js:13:12)

at Module._compile (internal/modules/cjs/loader.js:778:30)

at Object.Module._extensions..js (internal/modules/cjs/loader.js:789:10)

at Module.load (internal/modules/cjs/loader.js:653:32)

at tryModuleLoad (internal/modules/cjs/loader.js:593:12)

at Function.Module._load (internal/modules/cjs/loader.js:585:3)

at Module.require (internal/modules/cjs/loader.js:692:17)

i got this error…can anyone help me?

REPLY

desperate_student

August 20, 2020 at 9:18 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4409)

I found this solution and it worked for me :
https://stackoverflow.com/questions/50093144/mysql-8-0-client-does-not-support-
authentication-protocol-requested-by-server
(https://stackoverflow.com/questions/50093144/mysql-8-0-client-does-not-support-
authentication-protocol-requested-by-server)

REPLY

Philip

December 10, 2020 at 3:42 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5717)

go to MySQL Installer > Reconfigure MySQL Server > Authentication Method and select
"Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)"

We use cookies to improve your experience with the site. To find out more, you can read the full

REPLY

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

X

Vincent F

July 13, 2020 at 1:01 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3725)

Great tutorial and really appreciated. CRUD applications are real-world problems many tutorials can seem to overlook. This was perfect.

REPLY

Harry Karani

July 17, 2020 at 7:19 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3786)

Nice tutorial, especially for anyone doing an intro to REST APIs in NodeJS. Simple and straight to the point. Well done

REPLY

Shivani

July 25, 2020 at 11:04 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3940)

I have the following error. It will be a great help if you can it as soon as possible.
C:\Program Files\nodejs\nodejs-express-mysql>node server.js
Server is running on port 3000.
C:\Program Files\nodejs\nodejs-express-mysql\app\node_modules\mysql\lib\protocol\Parser.js:437
throw err; // Rethrow non-MySQL errors
^

Error: ER_ACCESS_DENIED_ERROR: Access denied for user 'root'@'localhost' (using password: YES)
at Handshake.Sequence._packetToError (C:\Program Files\nodejs\nodejs-express-mysql\app\node_modules\mysql\lib\protocol\sequences\Sequence.js:47:14)
at Handshake.ErrorPacket (C:\Program Files\nodejs\nodejs-express-mysql\app\node_modules\mysql\lib\protocol\sequences\Handshake.js:123:18)
at Protocol._parsePacket (C:\Program Files\nodejs\nodejs-express-mysql\app\node_modules\mysql\lib\protocol\Protocol.js:291:23)
at Parser._parsePacket (C:\Program Files\nodejs\nodejs-express-mysql\app\node_modules\mysql\lib\protocol\Parser.js:433:10)
at Parser.write (C:\Program Files\nodejs\nodejs-express-mysql\app\node_modules\mysql\lib\protocol\Parser.js:43:10)
at Protocol.write (C:\Program Files\nodejs\nodejs-express-

X

We use cookies to improve your experience with the site. To find out more, you can read the full **Privacy & Policy (https://bezkoder.com/privacy-policy/)**   Accept

mysql\app\node_modules\mysql\lib\protocol\Protocol.js:38:16)

at Socket. (C:\Program Files\nodejs\nodejs-express-

mysql\app\node_modules\mysql\lib\Connection.js:88:28)

at Socket. (C:\Program Files\nodejs\nodejs-express-

mysql\app\node_modules\mysql\lib\Connection.js:526:10)

at Socket.emit (events.js:315:20)

at addChunk (_stream_readable.js:295:12)

——————–

at Protocol._enqueue (C:\Program Files\nodejs\nodejs-express-

mysql\app\node_modules\mysql\lib\protocol\Protocol.js:144:48)

at Protocol.handshake (C:\Program Files\nodejs\nodejs-express-

mysql\app\node_modules\mysql\lib\protocol\Protocol.js:51:23)

at Connection.connect (C:\Program Files\nodejs\nodejs-express-

mysql\app\node_modules\mysql\lib\Connection.js:116:18)

at Object. (C:\Program Files\nodejs\nodejs-express-mysql\app\models\db.js:13:12)

at Module._compile (internal/modules/cjs/loader.js:1138:30)

at Object.Module._extensions..js (internal/modules/cjs/loader.js:1158:10)

at Module.load (internal/modules/cjs/loader.js:986:32)

at Function.Module._load (internal/modules/cjs/loader.js:879:14)

at Module.require (internal/modules/cjs/loader.js:1026:19)

at require (internal/modules/cjs/helpers.js:72:18) {

code: 'ER_ACCESS_DENIED_ERROR',

errno: 1045,

sqlMessage: "Access denied for user 'root'@'localhost' (using password: YES)",

sqlState: '28000',

fatal: true

}

REPLY

bezkoder

July 25, 2020 at 3:15 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3943)

Hi, you should create database with correct host, username, password, port. Then
modify the configuration in the source code.

REPLY

X

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)　　　Accept
Shivani

July 27, 2020 at 4:43 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3992)

also one query customers is one table here but in my case i have to join 3 tables and then put that data in kendro grid then how can i put that query in this code. please help.

how can i do it.

REPLY

Shivani

July 27, 2020 at 4:53 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3993)

And there are three join queries for each section on the front end. It will be a great help if you could provide entire solution as you have join on your website. Thanks in Advance.

REPLY

awanti

July 25, 2020 at 5:03 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-3949)

hello.......in my console it is just showing Server running on port 3000, but it is not showing Successfully connected database....and while sending requests on postman it shows that request cannot be send ....please give reply..please suggest something to work properly

REPLY

Shivani

July 30, 2020 at 2:53 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4070)

how to solve the following error . please help it is urgent.
events.js:292
throw er; // Unhandled 'error' event
^

Error: listen EADDRINUSE: address already in use :::3000

at Server.setupListenHandle [as _listen2] (net.js:1313:16)
at listenInCluster (net.js:1361:12)
at Server.listen (net.js:1447:7)

at Function.listen (C:\Program Files\nodejs\nodejs-express-
mysql\node_modules\express\lib\application.js:618:24)

at Object. (C:\Program Files\nodejs\nodejs-express-mysql\server.js:19:5)

at Module._compile (internal/modules/cjs/loader.js:1138:30)

at Object.Module._extensions..js (internal/modules/cjs/loader.js:1158:10)

at Module.load (internal/modules/cjs/loader.js:986:32)

at Function.Module._load (internal/modules/cjs/loader.js:879:14)

at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:71:12)

Emitted 'error' event on Server instance at:

at emitErrorNT (net.js:1340:8)

at processTicksAndRejections (internal/process/task_queues.js:84:21) {

code: 'EADDRINUSE',

errno: 'EADDRINUSE',

syscall: 'listen',

address: '::',

port: 3000

}

REPLY

bezkoder

July 30, 2020 at 2:55 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4071)

Hi, you need to stop the service which is using port 3000, or change the app server port.

REPLY

Shivani

July 30, 2020 at 11:41 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4084)

okay thank you so much

REPLY

Mapi

August 5, 2020 at 3:56 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4209)

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

Hi,

thank you for your clear and simple tutorial. Im new to javascript and node.js.

Can you explain why you didn't work with the ES6 syntax?

e.g. customer:

```
class Customer{
  constructor(name, email, active) {
    this.name= name;
    this.email= email;
    this.active=active;
  }
}
```

I know object oriented languages like Java – is your way the 'recommended' way in javascript? Or should I work with this ES6 syntax for better design?

Thank you

REPLY

Hunor

August 12, 2020 at 7:06 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4289)

I was wondering why the routing was implemented via passing the app object into the exported routing function instead of using the Router object that express provides.

Is it because it's one less line of code per route?

REPLY

Orholt

August 14, 2020 at 10:43 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4309)

Hi, I had problem with removing item from my MySQL DB, can someone help me?
I posted the issue on StackOverflow but only answer don't seems to work:
https://stackoverflow.com/questions/63400833/node-js-api-cant-delete-from-database
(https://stackoverflow.com/questions/63400833/node-js-api-cant-delete-from-database)

REPLY   ☒

Cristian

August 16, 2020 at 1:35 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4333)

Hello,

excellent article. It was very helpful to me. I have a question, because in the models, the errors are returned where the data should go. for example:

```
Customer.getAll = result => {
    sql.query ("SELECT * FROM customers", (err, res) => {
      if (err) {
        console.log ("error:", err);
        result (null, err);
        return;
      }

      console.log ("customers:", res);
      result (null, res);
    });
  };
```

Instead of:

```
  if (err) {
        result (err, null);
        return;
      }
```

Thanks for the article!

REPLY

Abel Heredia Bravo

August 16, 2020 at 7:54 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4338)

Excelent !!!!

REPLY

☒

We use cookies to improve your experience with the site. To find out more, you can read the full

Dragon
Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept
August 21, 2020 at 2:34 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4413)

WIth this controller-model solution, is it possible to handle async and wait process?

bezkoder

May 24, 2021 at 10:45 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-8690)

Hi, yes you can 🙂

It looks like:

```
exports.findAll = async (req, res, next) => {
  const { ok, result, error } = await Customer.getAll();
  if (!ok) {
    next(error)
  }
  return res.send(result);
};
```

Khashaa

August 27, 2020 at 7:11 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4486)

It was freakin helpful for me!

kodingspot

September 8, 2020 at 5:06 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4647)

is beautiful if you using new Promise() on the model….

[X]

REST
September 11, 2020 at 3:26 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4686)

This tutorial is incomplete. After you write:

"npm install express mysql body-parser –save"

You never generate any files. Therefore all the controller-files are missing. Other people have had problems with this in above comments. The app-folder is not present for example.

If you are going to make a step by step tutorial, please make it step by step and don't miss out on os-commands.

REPLY

kirad

September 15, 2020 at 1:53 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4726)

Great

REPLY

jahangir

October 7, 2020 at 11:41 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4944)

how to generate package-lock.json file ?

REPLY

bezkoder

October 7, 2020 at 1:40 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4945)

Hi, please run the command: `npm install`

REPLY

Bisht

October 8, 2020 at 6:48 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-4956)

Hello,

I am getting this error can you help.

TypeError: sql.query is not a function
at Function.Customer.create (/home/bisht/nodejs-express-
mysql/app/models/customer.model.js:11:7)
at exports.create (/home/bisht/nodejs-express-
mysql/app/controllers/customer.controller.js:20:14)
at Layer.handle [as handle_request] (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/layer.js:95:5)
at next (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/route.js:137:13)
at Route.dispatch (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/route.js:112:3)
at Layer.handle [as handle_request] (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/layer.js:95:5)
at /home/bisht/nodejs-express-mysql/node_modules/express/lib/router/index.js:281:22
at Function.process_params (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/index.js:335:12)
at next (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/index.js:275:10)
at urlencodedParser (/home/bisht/nodejs-express-mysql/node_modules/body-
parser/lib/types/urlencoded.js:82:7)
TypeError: sql.query is not a function
at Function.Customer.getAll (/home/bisht/nodejs-express-
mysql/app/models/customer.model.js:43:7)
at exports.findAll (/home/bisht/nodejs-express-
mysql/app/controllers/customer.controller.js:32:14)
at Layer.handle [as handle_request] (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/layer.js:95:5)
at next (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/route.js:137:13)
at Route.dispatch (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/route.js:112:3)
at Layer.handle [as handle_request] (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/layer.js:95:5)
at /home/bisht/nodejs-express-mysql/node_modules/express/lib/router/index.js:281:22
at Function.process_params (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/index.js:335:12)
at next (/home/bisht/nodejs-express-
mysql/node_modules/express/lib/router/index.js:275:10)
at urlencodedParser (/home/bisht/nodejs-express-mysql/node_modules/body-
parser/lib/types/urlencoded.js:82:7)

Ghanshyam Sharma
November 2, 2020 at 8:17 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5264)

Hello,
For this you need to add sql module *npm install mysql* and after include *var mysql =
require('mysql');*
in your server.js or index.js

Thanks

REPLY

Aviroop Sanyal
November 12, 2020 at 6:07 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5375)

Thanks for your post. It was really helpful. Please help me with customer login with your
rest api format

REPLY

bezkoder
November 12, 2020 at 8:20 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5376)

Hi, you can find the instruction at:
Node.js Express: JWT example | Token Based Authentication & Authorization
(https://bezkoder.com/node-js-jwt-authentication-mysql/)

REPLY

Manasa
November 19, 2020 at 6:46 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5461)

Thank you.Helped me a lot as a beginner.

REPLY

X

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

Taufik Muharrom
December 11, 2020 at 7:18 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5724)

are "insertId" and "affectedRows" a method from sql ? i am still confused , where did the method come from.

REPLY

bezkoder
December 11, 2020 at 2:13 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5731)

Hi, they came from response data of MySQL query 😊

REPLY

Ajay
December 28, 2020 at 6:05 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-5945)

thank you

REPLY

Shubham Verma
January 4, 2021 at 2:02 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-6072)

Thank You so much... (Very neat and precise explanation)

REPLY

mugesh
January 30, 2021 at 7:48 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-6694)

postman error bro

{

"message": "ER_BAD_NULL_ERROR: Column 'content' cannot be null"

}

error status: 500 internet server error

REPLY

bezkoder

May 24, 2021 at 10:47 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-8691)

Hi, kindly make sure that you have `Content-Type: application/json` in the HTTP request header 🙂

REPLY

Raj

February 16, 2021 at 6:32 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-7056)

works like a charm, well explained and thanks for sharing .....keep up the good work and hope to see of more tutorials using react + nodejs, angular + nodejs with real-world examples and with authentication + authorization

REPLY

Minerva

March 9, 2021 at 3:36 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-7499)

Nice tutorial, thanks.
Just FYI, small bug/typo in model section:
Should be "result(err, null)" instead of "result(null, err)"

REPLY

loeten

August 18, 2021 at 8:31 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-10991)

This is really a great tutorial! But indeed, only with

result(err, null);

I get a correct (wanted) error response when trying to remove all customers; I did set a
We use cookies to improve your experience with the site. To find out more, you can read the full
constraint in the database, where the customer id is a foreign key in another table with
valid data. So it is not possible in the DB to delete the (referenced) customer only. But

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

X

the API reports "all customers deleted successfully" 😊 until I changed this line as you suggested.

REPLY

aaa

April 3, 2021 at 11:10 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-7877)

hey hope you having a great day where should we implement the functions please ?

REPLY

Ashik Ali Shanto

April 9, 2021 at 12:28 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-8020)

{"message":"ER_BAD_NULL_ERROR: Column 'email' cannot be null"}
while post using postman

REPLY

Gwen

April 10, 2021 at 5:10 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-8034)

Thank you very much. It's one of the most clear tutorial on this subject I ever found.
So would you have an advice on an article clearly explaining how to add a security layer on this, like with token ….

TY

REPLY

Vignesh

April 18, 2021 at 3:41 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-8151)

I saw your sequalize tutorial very nice, I want to do the same in this example, I mean joining multiple table, how to achieve the association concept here, I know I can use sequalize but I want understand more in the boilerplate example

REPLY

Vignesh

April 18, 2021 at 4:05 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-8153)

How to achieve multiple table join in this example, I know we can use sequalize but I want to understand more in this example

REPLY

Reece

April 30, 2021 at 12:52 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-8323)

Great stuff.
Best tutorial I've found on the topic.

REPLY

shaikh

May 6, 2021 at 12:25 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-8408)

Very helpful!
Thank you so much man!

REPLY

vineet

June 10, 2021 at 12:54 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-9003)

when i am accesing this url http://localhost:8000/customers (http://localhost:8000/customers) i am getting this message –
Cannot GET /customers

REPLY

[X]

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

vineet

June 10, 2021 at 1:57 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-9004)

The problem that i posted few minute ago "can not get / customers" is resolved but i am facing other problem when i request URL in postman only sending request is coming no response is return so please help me to resolve this issue

REPLY

Toyyib
June 24, 2021 at 1:22 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-9265)

require("../node_mysql/routes/customer.routes.js")(app)

What does this line do?

REPLY

Raj Rajput
July 15, 2021 at 3:44 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-9607)

it's a very helpful article.
I request you please create another article for user registration and login with express validator & Token in express with MySQL.

Thanks

REPLY

bezkoder
July 16, 2021 at 1:49 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-9630)

Hi, thanks for your suggestion.
We have the tutorial at: Node.js Express: JWT example | Token Based Authentication & Authorization (https://www.bezkoder.com/node-js-jwt-authentication-mysql/)

REPLY

Jatal Madhaji
July 23, 2021 at 11:07 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-9923)

Thank you man .. Well explained

REPLY

MOHAMED ZAKRIA AL MOKDAD

July 24, 2021 at 1:27 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-9963)

merci bien

REPLY

bruno costa

July 25, 2021 at 3:40 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-10009)

bodyparser is deprecated, use express.json and express.urlencoded instead

REPLY

Max Delgado

August 13, 2021 at 3:00 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-10796)

Greetings! I am getting the error:
{
code: 'ER_NOT_SUPPORTED_AUTH_MODE',
errno: 1251,
sqlMessage: 'Client does not support authentication protocol requested by server;
consider upgrading MySql client',
sqlState: '08004',
fatal: true
}

REPLY

Ankit Bhadana

September 29, 2021 at 11:28 am (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-12799)

Amazing stuff! I found it all really helpful!

X

REPLY

Shahrizal

September 29, 2021 at 2:43 pm (https://www.bezkoder.com/node-js-rest-api-express-mysql/#comment-12810)

Thanks a lot. It's helps me a lot to start with my node js express journey

REPLY

Leave a Reply

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

Comment

Name *

Email *

Website

☐

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

❮ React Hooks File Upload example with Axios & Progress Bar (https://www.bezkoder.com/react-hooks-file-upload/)

Search…   🔍

FOLLOW US   ☒

We use cookies to improve your experience with the site. To find out more, you can read the full

Privacy & Policy (https://bezkoder.com/privacy-policy/)   Accept

▶

(htt

ps://

ww

w.yo

utub

e.co

m/c

han

𝐟 nel/ ◯

(htt UCp (htt

ps:// 0mx ps://

face 9RH gith

boo 0Jxa ub.c

k.co Fsm om/

m/b MvK bezk

ezko XA8 oder

der) 6Q))

TOOLS

Json Formatter (https://www.bezkoder.com/json-formatter/)

Home (https://bezkoder.com/)   Privacy Policy (https://www.bezkoder.com/privacy-policy/)

Contact Us (https://www.bezkoder.com/contact-us/)   About Us (https://www.bezkoder.com/about/)

BezKoder 2019