Node.js Express: JWT example | Token Based Authentication & Authorization

Last modified: August 17, 2021 (https://www.bezkoder.com/node-js-jwt-authentication-mysql/) bezkoder (https://www.bezkoder.com/author/bezkoder/) Node.js (https://www.bezkoder.com/category/node-js/), Security (https://www.bezkoder.com/category/security/)

In this tutorial, we're gonna build a Node.js Express Rest API example that supports Token Based Authentication with JWT (JSONWebToken (https://www.npmjs.com/package/jsonwebtoken)). You'll know:

- Appropriate Flow for User Signup & User Login with JWT Authentication
- Node.js Express Architecture with CORS, Authenticaton & Authorization middlewares & Sequelize
- How to configure Express routes to work with JWT
- How to define Data Models and association for Authentication and Authorization
- Way to use Sequelize to interact with MySQL Database

Related Posts:
– Node.js Rest APIs example with Express, Sequelize & MySQL (https://bezkoder.com/node-js-express-sequelize-mysql/)
– Node.js + MongoDB: User Authentication & Authorization with JWT (https://bezkoder.com/node-js-mongodb-auth-jwt/)
– Node.js + PostgreSQL: User Authentication & Authorization with JWT (https://bezkoder.com/node-js-jwt-authentication-postgresql/)

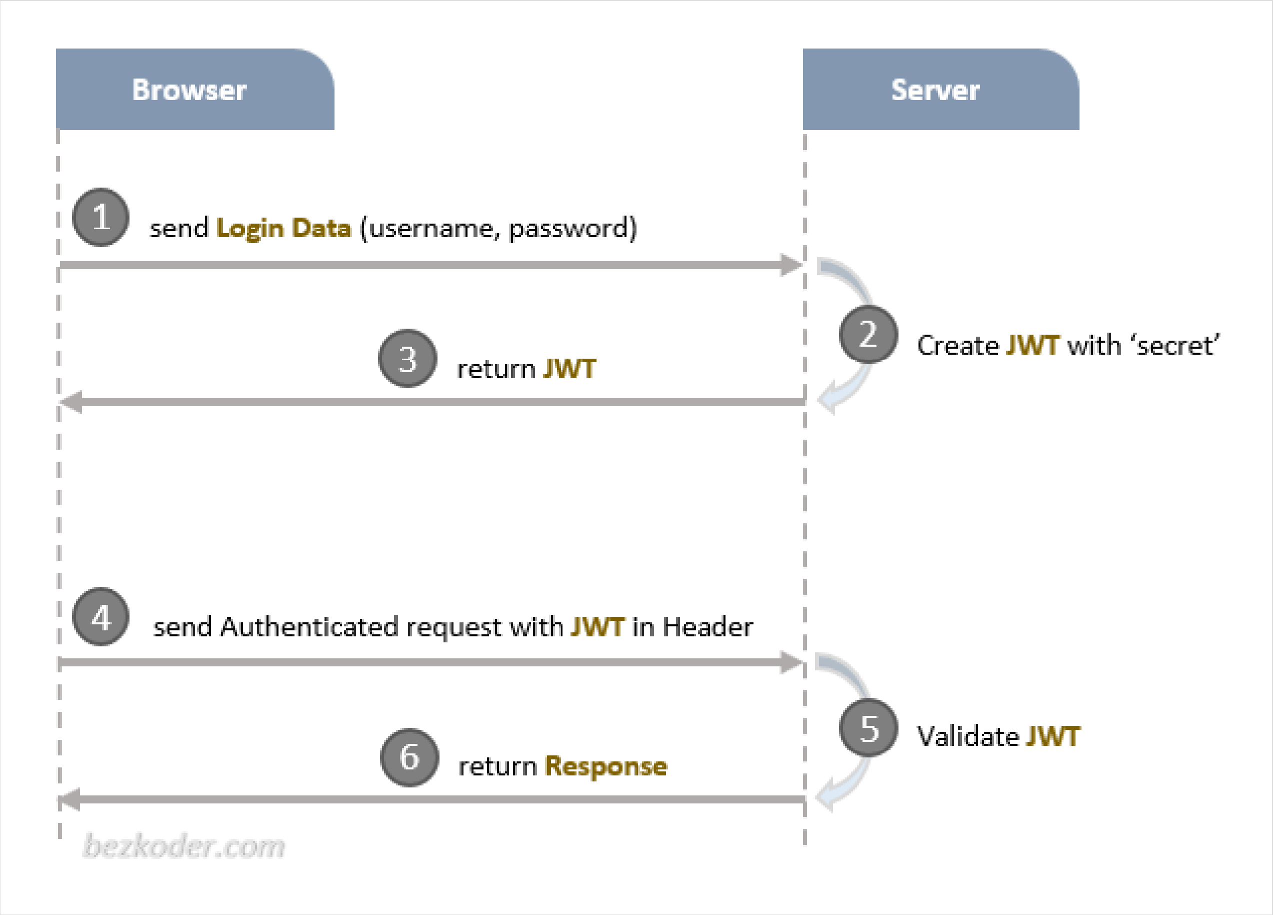
Fullstack (JWT Authentication & Authorization example):
– Node.js Express + Vue.js (https://bezkoder.com/node-express-vue-jwt-auth/)
– Node.js Express + Angular 8 (https://bezkoder.com/node-js-express-angular-jwt-auth/)
– Node.js Express + Angular 10 (https://bezkoder.com/node-js-express-angular-10-jwt-auth/)
– Node.js Express + Angular 11 (https://bezkoder.com/node-js-angular-11-jwt-authentication/)
– Node.js Express + Angular 12 (https://www.bezkoder.com/node-js-angular-12-jwt-auth/)
– Node.js Express + React (https://bezkoder.com/react-express-authentication-jwt/)

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/)

Contents [hide]

Token Based Authentication

Comparing with Session-based Authentication that need to store Session on Cookie, the big advantage of Token-based Authentication is that we store the JSON Web Token (JWT) on Client side: Local Storage for Browser, Keychain for IOS and SharedPreferences for Android… So we don’t need to build another backend project that supports Native Apps or an additional Authentication module for Native App users.


Browser
Server
1 send Login Data (username, password)
2 Create JWT with 'secret'
3 return JWT
4 send Authenticated request with JWT in Header
5 Validate JWT
6 return Response
bezkoder.com

There are three important parts of a JWT: Header, Payload, Signature. Together they are combined to a standard structure: `header.payload.signature` .

The Client typically attaches JWT in **Authorization** header with Bearer prefix:

```
Authorization: Bearer [header].[payload].[signature]
```

Or only in **x-access-token** header:

```
x-access-token: [header].[payload].[signature]
```

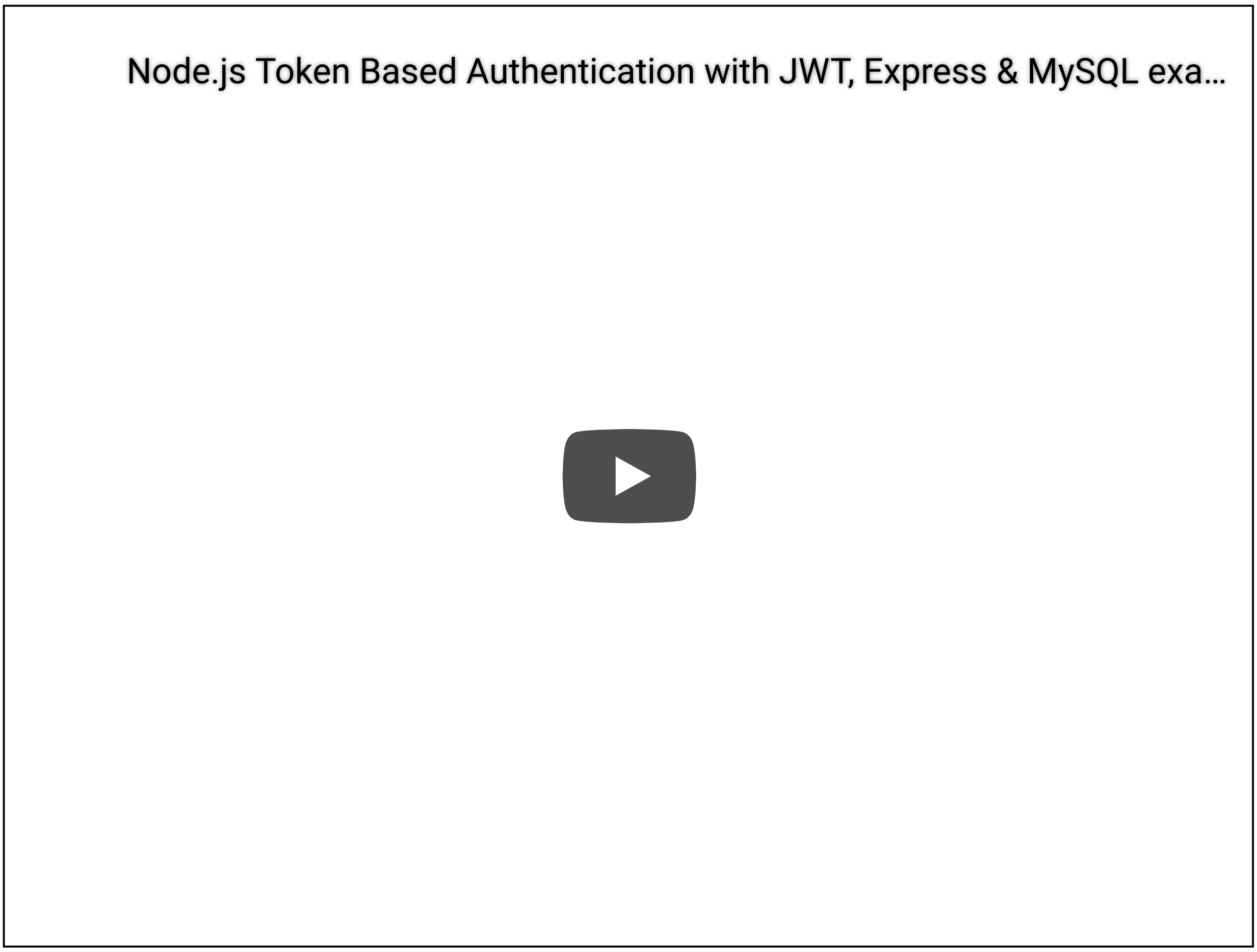
For more details, you can visit:
In-depth Introduction to JWT-JSON Web Token (https://bezkoder.com/jwt-json-web-token/)

Overview of Node.js Express JWT Authentication example

We will build a Node.js Express application in that:

- User can signup new account, or login with username & password.
- By User's role (admin, moderator, user), we authorize the User to access resources

This is our Node.js application demo running with MySQL database and test Rest Apis with Postman.

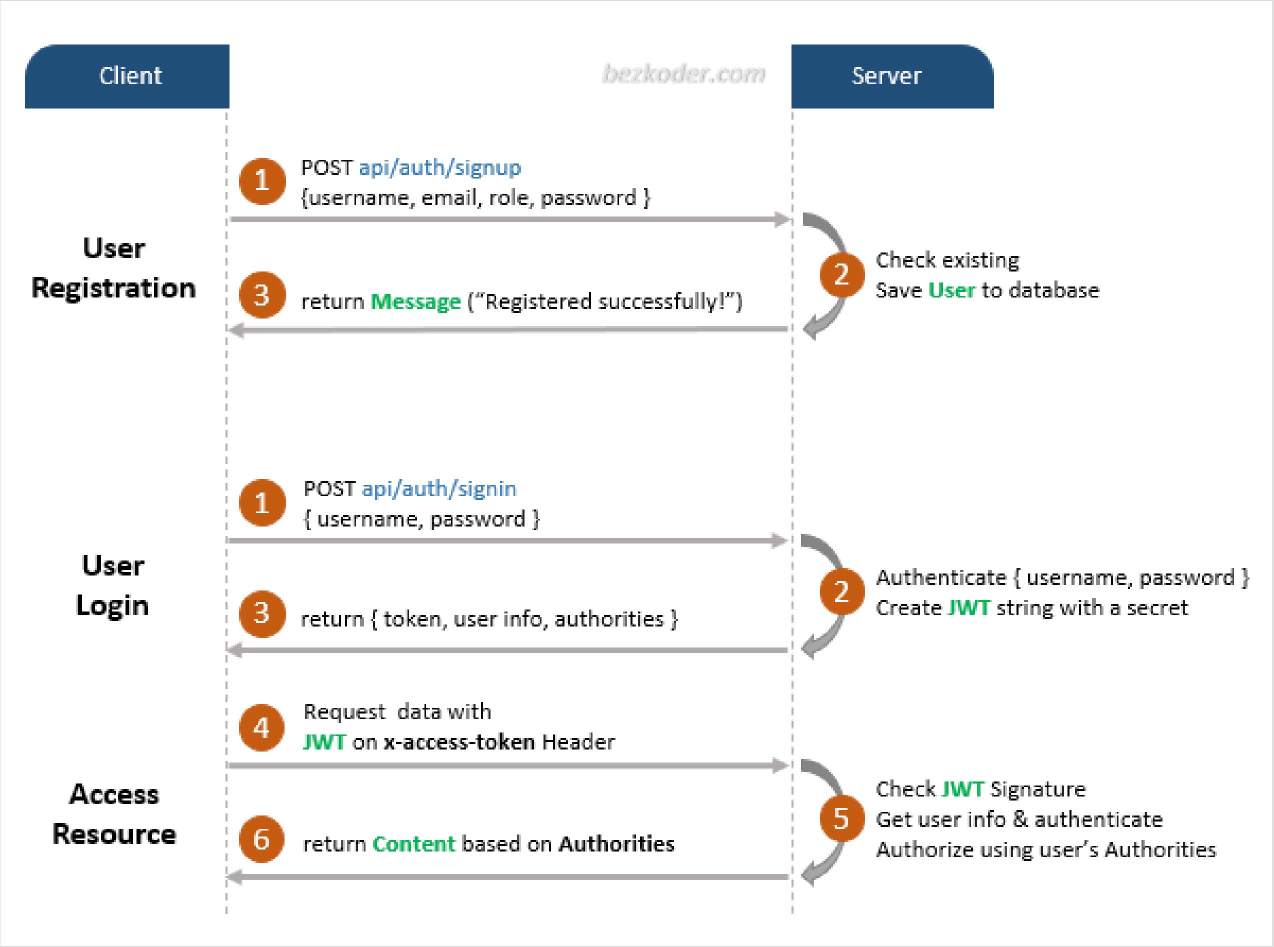

Node.js Token Based Authentication with JWT, Express & MySQL exa…

These are APIs that we need to provide:

Methods	Urls	Actions
POST	/api/auth/signup	signup new account
POST	/api/auth/signin	login an account
GET	/api/test/all	retrieve public content
GET	/api/test/user	access User’s content
GET	/api/test/mod	access Moderator’s content
GET	/api/test/admin	access Admin’s content

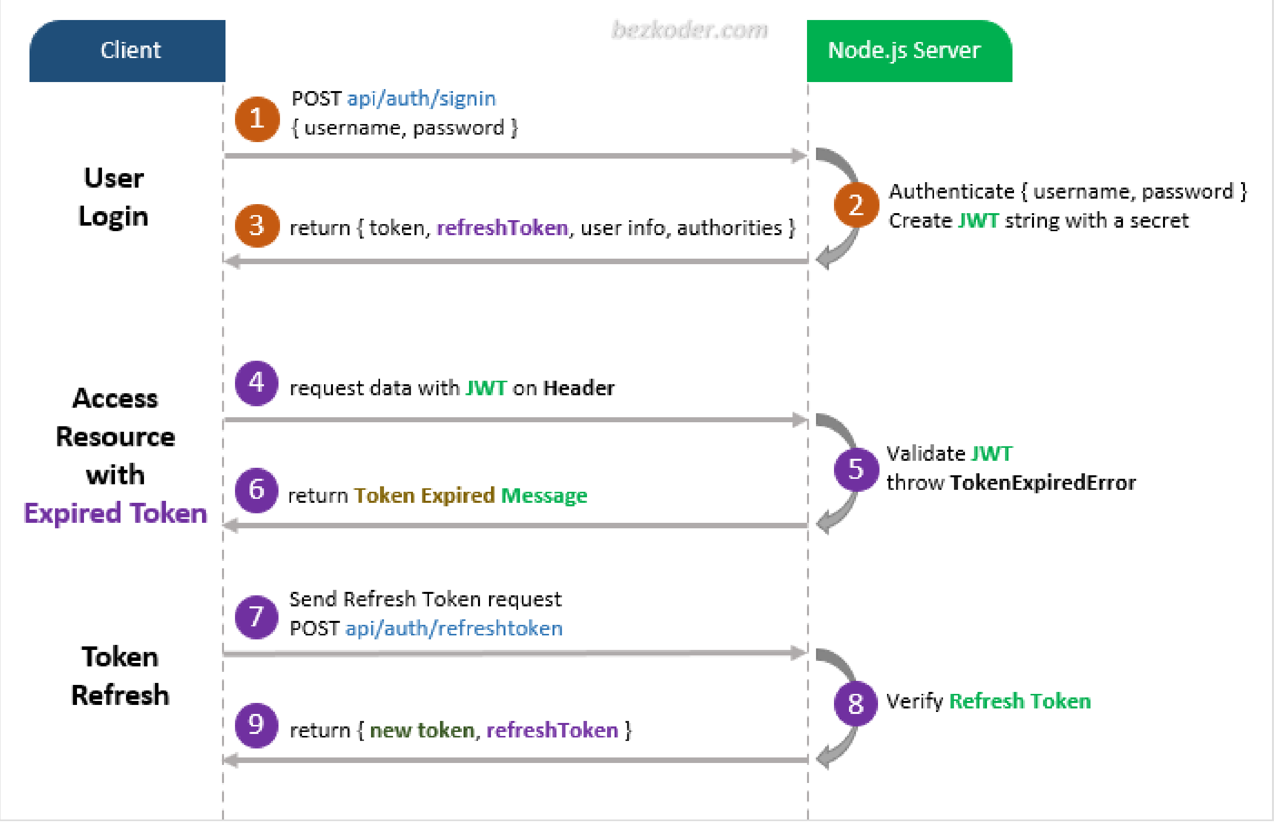
Flow for Signup & Login with JWT Authentication

The diagram shows flow of User Registration, User Login and Authorization process.


Client
bezkoder.com
Server
User Registration
1 POST api/auth/signup {username, email, role, password }
2 Check existing Save User to database
3 return Message ("Registered successfully!")
User Login
1 POST api/auth/signin { username, password }
2 Authenticate { username, password } Create JWT string with a secret
3 return { token, user info, authorities }
Access Resource
4 Request data with JWT on x-access-token Header
5 Check JWT Signature Get user info & authenticate Authorize using user's Authorities
6 return Content based on Authorities

A legal JWT must be added to HTTP **x-access-token** Header if Client accesses protected resources.

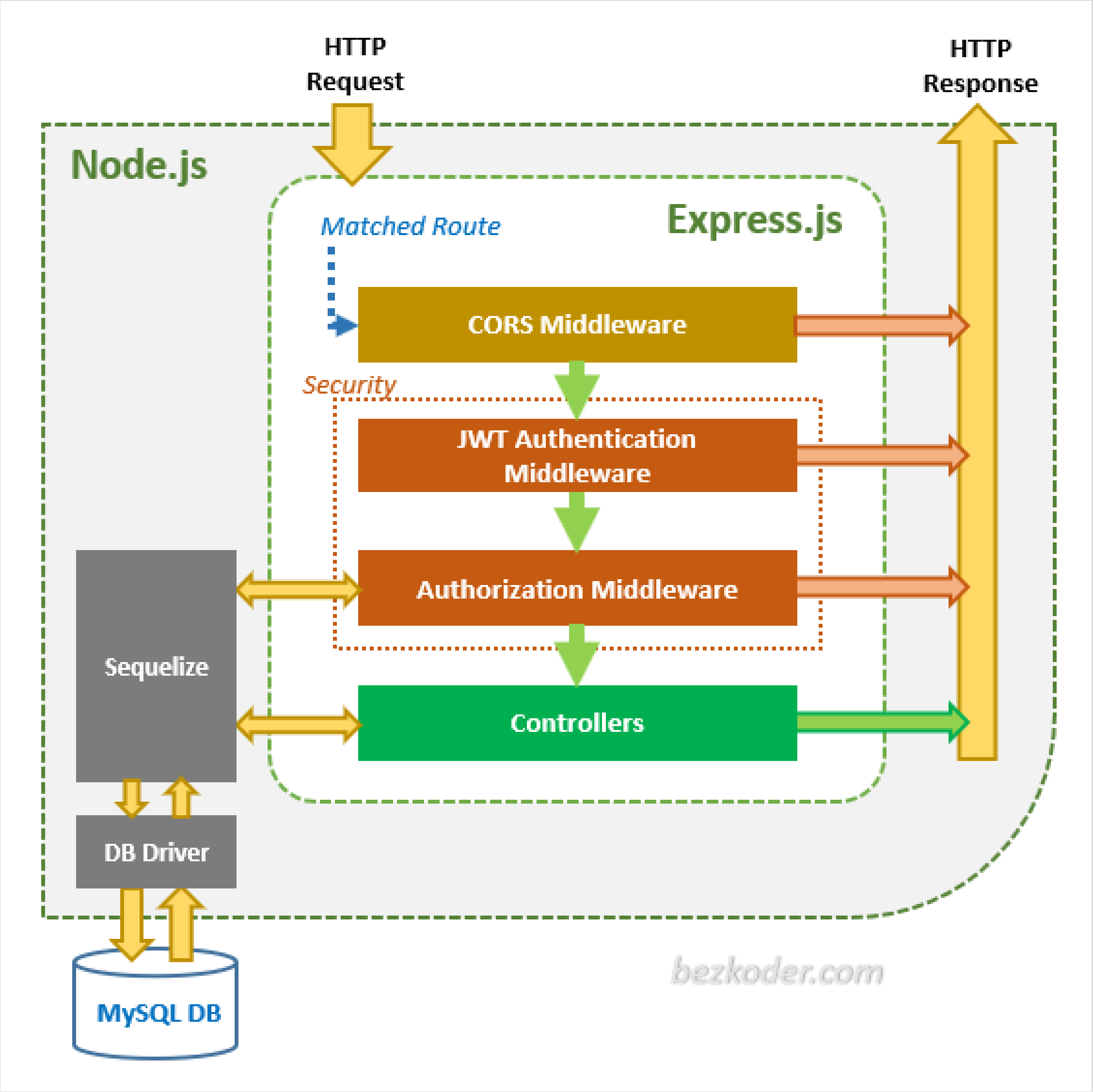
You will need to implement Refresh Token:


Client
bezkoder.com
Node.js Server
User Login
1 POST api/auth/signin { username, password }
2 Authenticate { username, password } Create JWT string with a secret
3 return { token, refreshToken, user info, authorities }
Access Resource with Expired Token
4 request data with JWT on Header
5 Validate JWT throw TokenExpiredError
6 return Token Expired Message
Token Refresh
7 Send Refresh Token request POST api/auth/refreshtoken
8 Verify Refresh Token
9 return { new token, refreshToken }

More details at: JWT Refresh Token implementation in Node.js example (https://bezkoder.com/jwt-refresh-token-node-js/)

Node.js Express Architecture with Authentication & Authorization

You can have an overview of our Node.js Express App with the diagram below:



Via *Express* routes, **HTTP request** that matches a route will be checked by **CORS Middleware** before coming to **Security** layer.

Security layer includes:

- JWT Authentication Middleware: verify SignUp, verify token
- Authorization Middleware: check User's roles with record in database

If these middlewares throw any error, a message will be sent as HTTP response.

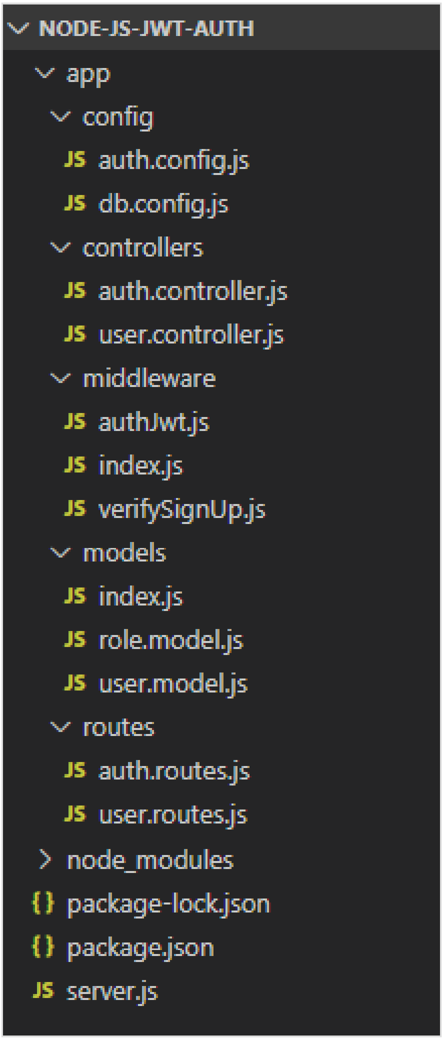
Controllers interact with MySQL Database via *Sequelize* and send **HTTP response** (token, user information, data based on roles…) to client.

Technology

- Express 4.17.1
- bcryptjs 2.4.3
- jsonwebtoken 8.5.1
- Sequelize 5.21.3
- MySQL

Project Structure

This is directory structure for our Node.js Express application:



```
NODE-JS-JWT-AUTH
  app
    config
      auth.config.js
      db.config.js
    controllers
      auth.controller.js
      user.controller.js
    middleware
      authJwt.js
      index.js
      verifySignUp.js
    models
      index.js
      role.model.js
      user.model.js
    routes
      auth.routes.js
      user.routes.js
  node_modules
  package-lock.json
  package.json
  server.js
```

– **config**

- configure MySQL database & Sequelize
- configure Auth Key

– **routes**

- *auth.routes.js*: POST signup & signin
- *user.routes.js*: GET public & protected resources

– **middlewares**

- *verifySignUp.js*: check duplicate Username or Email
- *authJwt.js*: verify Token, check User roles in database

– **controllers**

- *auth.controller.js*: handle signup & signin actions
- *user.controller.js*: return public & protected content

– **models** for Sequelize Models

- *user.model.js*
- *role.model.js*

– *server.js*: import and initialize neccesary modules and routes, listen for connections.

Create Node.js App

First, we create a folder for our project:

```
$ mkdir node-js-jwt-auth
$ cd node-js-jwt-auth
```

Then we initialize the Node.js App with a *package.json* file:

```
npm init

name: (node-js-jwt-auth)
version: (1.0.0)
description: Node.js Demo for JWT Authentication
entry point: (index.js) server.js
test command:
git repository:
keywords: node.js, express, jwt, authentication, mysql
author: bezkoder
license: (ISC)

Is this ok? (yes) yes
```

We need to install necessary modules: `express` , `cors` , `body-parser` , `sequelize` , `mysql2` , `jsonwebtoken` and `bcryptjs` .
Run the command:

```
npm install express sequelize mysql2 body-parser cors jsonwebtoken bcryptjs --
save
```

The package.json file now looks like this:

```
{
  "name": "node-js-jwt-auth",
  "version": "1.0.0",
  "description": "Node.js Demo for JWT Authentication",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "node.js",
    "jwt",
    "authentication",
    "express",
    "mysql"
  ],
  "author": "bezkoder",
  "license": "ISC",
  "dependencies": {
    "bcryptjs": "^2.4.3",
    "body-parser": "^1.19.0",
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "jsonwebtoken": "^8.5.1",
    "mysql2": "^2.1.0",
    "sequelize": "^5.21.3"
  }
}
```

Setup Express web server

In the root folder, let’s create a new *server.js* file:

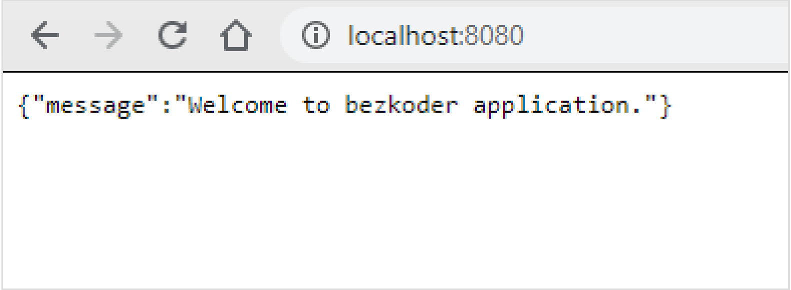
```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");

const app = express();

var corsOptions = {
  origin: "http://localhost:8081"
};

app.use(cors(corsOptions));

// 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
const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}.`);
});
```

Let me explain what we've just done:
– import `express` , `body-parser` and `cors` modules:

- Express is for building the Rest apis
- body-parser (https://www.npmjs.com/package/body-parser) helps to parse the request and create the `req.body` object
- cors (https://www.npmjs.com/package/cors) provides Express middleware to enable CORS

– create an Express app, then add `body-parser` and `cors` middlewares using `app.use()` method. Notice that we set origin: `http://localhost:8081` .
– define a GET route which is simple for test.
– listen on port 8080 for incoming requests.

Now let's run the app with command: `node server.js` .
Open your browser with url http://localhost:8080/ (http://localhost:8080/), you will see:



```
{"message":"Welcome to bezkoder application."}
```

Configure MySQL database & Sequelize

In the **app** folder, create **config** folder for configuration with *db.config.js* file like this:

```
module.exports = {
  HOST: "localhost",
  USER: "root",
  PASSWORD: "123456",
  DB: "testdb",
  dialect: "mysql",
  pool: {
    max: 5,
    min: 0,
    acquire: 30000,
    idle: 10000
  }
};
```

First five parameters are for MySQL connection.
`pool` is optional, it will be used for Sequelize connection pool configuration:

- `max` : maximum number of connection in pool
- `min` : minimum number of connection in pool
- `idle` : maximum time, in milliseconds, that a connection can be idle before being released
- `acquire` : maximum time, in milliseconds, that pool will try to get connection before throwing error

For more details, please visit API Reference for the Sequelize constructor (https://sequelize.org/master/class/lib/sequelize.js~Sequelize.html#instance-constructor-constructor).

Define the Sequelize Model

In *models* folder, create `User` and `Role` data model as following code:

models/*user.model.js*

```
module.exports = (sequelize, Sequelize) => {
  const User = sequelize.define("users", {
    username: {
      type: Sequelize.STRING
    },
    email: {
      type: Sequelize.STRING
    },
    password: {
      type: Sequelize.STRING
    }
  });

  return User;
};
```

models/*role.model.js*

```
module.exports = (sequelize, Sequelize) => {
  const Role = sequelize.define("roles", {
    id: {
      type: Sequelize.INTEGER,
      primaryKey: true
    },
    name: {
      type: Sequelize.STRING
    }
  });

  return Role;
};
```

These Sequelize Models represents **users** & **roles** table in MySQL database.

After initializing Sequelize, we don’t need to write CRUD functions, Sequelize supports all of them:

- create a new User: `create (https://sequelize.org/master/class/lib/model.js~Model.html#static-method-create)(object)`
- find a User by id: `findByPk (https://sequelize.org/master/class/lib/model.js~Model.html#static-method-findByPk)(id)`
- find a User by email: `findOne (https://sequelize.org/master/class/lib/model.js~Model.html#static-method-findOne)({ where: { email: ... } })`

- get all Users: `findAll (https://sequelize.org/master/class/lib/model.js~Model.html#static-method-findAll)()`
- find all Users by *username*: `findAll({ where: { username: ... } })`

These functions will be used in our Controllers and Middlewares.

Initialize Sequelize

Now create **app**/**models**/*index.js* with content like this:

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

const Sequelize = require("sequelize");
const sequelize = new Sequelize(
  config.DB,
  config.USER,
  config.PASSWORD,
  {
    host: config.HOST,
    dialect: config.dialect,
    operatorsAliases: false,

    pool: {
      max: config.pool.max,
      min: config.pool.min,
      acquire: config.pool.acquire,
      idle: config.pool.idle
    }
  }
);

const db = {};

db.Sequelize = Sequelize;
db.sequelize = sequelize;

db.user = require("../models/user.model.js")(sequelize, Sequelize);
db.role = require("../models/role.model.js")(sequelize, Sequelize);

db.role.belongsToMany(db.user, {
  through: "user_roles",
  foreignKey: "roleId",
  otherKey: "userId"
});
db.user.belongsToMany(db.role, {
  through: "user_roles",
  foreignKey: "userId",
  otherKey: "roleId"
});

db.ROLES = ["user", "admin", "moderator"];

module.exports = db;
```

The association between *Users* and *Roles* is Many-to-Many relationship:
– One User can have several Roles.
– One Role can be taken on by many Users.

We use `User.belongsToMany(Role)` to indicate that the *user* model can belong to many *Role*s and vice versa.

With `through, foreignKey, otherKey`, we're gonna have a new table **user_roles** as connection between **users** and **roles** table via their primary key as foreign keys.

If you want to know more details about how to make Many-to-Many Association with Sequelize and Node.js, please visit:
Sequelize Many-to-Many Association example – Node.js & MySQL (https://bezkoder.com/sequelize-associate-many-to-many/)

Don't forget to call `sync()` method in *server.js*.

```
...
const app = express();
app.use(...);

const db = require("./app/models");
const Role = db.role;

db.sequelize.sync({force: true}).then(() => {
  console.log('Drop and Resync Db');
  initial();
});

...
function initial() {
  Role.create({
    id: 1,
    name: "user"
  });

  Role.create({
    id: 2,
    name: "moderator"
  });

  Role.create({
    id: 3,
    name: "admin"
  });
}
```

`initial()` function helps us to create 3 rows in database.
In development, you may need to drop existing tables and re-sync database. So you can use `force: true` as code above.

For production, just insert these rows manually and use `sync()` without parameters to avoid dropping data:

```
...
const app = express();
app.use(...);

const db = require("./app/models");

db.sequelize.sync();
...
```

Learn how to implement Sequelize One-to-Many Relationship at:
Sequelize Associations: One-to-Many example – Node.js, MySQL (https://bezkoder.com/sequelize-associate-one-to-many/)

Configure Auth Key

jsonwebtoken functions such as `verify()` or `sign()` use algorithm that needs a secret key (as String) to encode and decode token.

In the **app**/**config** folder, create *auth.config.js* file with following code:

```
module.exports = {
  secret: "bezkoder-secret-key"
};
```

You can create your own `secret` String.

Create Middleware functions

To verify a Signup action, we need 2 functions:
– check if `username` or `email` is duplicate or not
– check if `roles` in the request is existed or not

middleware/*verifySignUp.js*

```
const db = require("../models");
const ROLES = db.ROLES;
const User = db.user;

checkDuplicateUsernameOrEmail = (req, res, next) => {
  // Username
  User.findOne({
    where: {
      username: req.body.username
    }
  }).then(user => {
    if (user) {
      res.status(400).send({
        message: "Failed! Username is already in use!"
      });
      return;
    }

    // Email
    User.findOne({
      where: {
        email: req.body.email
      }
    }).then(user => {
      if (user) {
        res.status(400).send({
          message: "Failed! Email is already in use!"
        });
        return;
      }

      next();
    });
  });
};

checkRolesExisted = (req, res, next) => {
  if (req.body.roles) {
    for (let i = 0; i < req.body.roles.length; i++) {
      if (!ROLES.includes(req.body.roles[i])) {
        res.status(400).send({
          message: "Failed! Role does not exist = " + req.body.roles[i]
        });
        return;
      }
    }
  }

  next();
};

const verifySignUp = {
  checkDuplicateUsernameOrEmail: checkDuplicateUsernameOrEmail,
  checkRolesExisted: checkRolesExisted
};

module.exports = verifySignUp;
```

To process Authentication & Authorization, we have these functions:
- check if `token` is provided, legal or not. We get token from **x-access-token** of HTTP headers, then use **jsonwebtoken**'s `verify()` function.
- check if `roles` of the user contains required role or not.

middleware/*authJwt.js*

```
const jwt = require("jsonwebtoken");
const config = require("../config/auth.config.js");
const db = require("../models");
const User = db.user;

verifyToken = (req, res, next) => {
  let token = req.headers["x-access-token"];

  if (!token) {
    return res.status(403).send({
      message: "No token provided!"
    });
  }

  jwt.verify(token, config.secret, (err, decoded) => {
    if (err) {
      return res.status(401).send({
        message: "Unauthorized!"
      });
    }
    req.userId = decoded.id;
    next();
  });
};

isAdmin = (req, res, next) => {
  User.findByPk(req.userId).then(user => {
    user.getRoles().then(roles => {
      for (let i = 0; i < roles.length; i++) {
        if (roles[i].name === "admin") {
          next();
          return;
        }
      }

      res.status(403).send({
        message: "Require Admin Role!"
      });
      return;
    });
  });
};

isModerator = (req, res, next) => {
  User.findByPk(req.userId).then(user => {
    user.getRoles().then(roles => {
      for (let i = 0; i < roles.length; i++) {
        if (roles[i].name === "moderator") {
          next();
          return;
        }
      }

      res.status(403).send({
        message: "Require Moderator Role!"
      });
    });
  });
};

isModeratorOrAdmin = (req, res, next) => {
  User.findByPk(req.userId).then(user => {
    user.getRoles().then(roles => {
      for (let i = 0; i < roles.length; i++) {
        if (roles[i].name === "moderator") {
          next();
          return;
        }

        if (roles[i].name === "admin") {
          next();
          return;
        }
      }

      res.status(403).send({
        message: "Require Moderator or Admin Role!"
      });
    });
  });
};

const authJwt = {
  verifyToken: verifyToken,
  isAdmin: isAdmin,
  isModerator: isModerator,
  isModeratorOrAdmin: isModeratorOrAdmin
};
module.exports = authJwt;
```

middleware/*index.js*

```
const authJwt = require("./authJwt");
const verifySignUp = require("./verifySignUp");

module.exports = {
  authJwt,
  verifySignUp
};
```

Create Controllers

Controller for Authentication

There are 2 main functions for Authentication:
- `signup` : create new User in database (role is **user** if not specifying role)
- `signin` :

- find `username` of the request in database, if it exists
- compare `password` with `password` in database using **bcrypt**, if it is correct
- generate a token using **jsonwebtoken**
- return user information & access Token

controllers/*auth.controller.js*

```
const db = require("../models");
const config = require("../config/auth.config");
const User = db.user;
const Role = db.role;

const Op = db.Sequelize.Op;

var jwt = require("jsonwebtoken");
var bcrypt = require("bcryptjs");

exports.signup = (req, res) => {
  // Save User to Database
  User.create({
    username: req.body.username,
    email: req.body.email,
    password: bcrypt.hashSync(req.body.password, 8)
  })
    .then(user => {
      if (req.body.roles) {
        Role.findAll({
          where: {
            name: {
              [Op.or]: req.body.roles
            }
          }
        }).then(roles => {
          user.setRoles(roles).then(() => {
            res.send({ message: "User was registered successfully!" });
          });
        });
      } else {
        // user role = 1
        user.setRoles([1]).then(() => {
          res.send({ message: "User was registered successfully!" });
        });
      }
    })
    .catch(err => {
      res.status(500).send({ message: err.message });
    });
};

exports.signin = (req, res) => {
  User.findOne({
    where: {
      username: req.body.username
    }
  })
    .then(user => {
      if (!user) {
        return res.status(404).send({ message: "User Not found." });
      }

      var passwordIsValid = bcrypt.compareSync(
        req.body.password,
        user.password
      );

      if (!passwordIsValid) {
        return res.status(401).send({
          accessToken: null,
          message: "Invalid Password!"
        });
      }

      var token = jwt.sign({ id: user.id }, config.secret, {
        expiresIn: 86400 // 24 hours
      });

      var authorities = [];
      user.getRoles().then(roles => {
        for (let i = 0; i < roles.length; i++) {
          authorities.push("ROLE_" + roles[i].name.toUpperCase());
        }
        res.status(200).send({
          id: user.id,
          username: user.username,
          email: user.email,
          roles: authorities,
          accessToken: token
        });
      });
    })
    .catch(err => {
      res.status(500).send({ message: err.message });
    });
};
```

Controller for testing Authorization

There are 4 functions:

– `/api/test/all` for public access

– `/api/test/user` for loggedin users (role: **user**/**moderator**/**admin**)

– `/api/test/mod` for users having **moderator** role

– `/api/test/admin` for users having **admin** role

controllers/*user.controller.js*

```
exports.allAccess = (req, res) => {
  res.status(200).send("Public Content.");
};

exports.userBoard = (req, res) => {
  res.status(200).send("User Content.");
};

exports.adminBoard = (req, res) => {
  res.status(200).send("Admin Content.");
};

exports.moderatorBoard = (req, res) => {
  res.status(200).send("Moderator Content.");
};
```

Now, do you have any question? Would you like to know how we can combine middlewares with controller functions?
Let's do it in the next section.

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 response by setting up the routes.

We can separate our routes into 2 part: for Authentication and for Authorization (accessing protected resources).

Authentication:

- POST `/api/auth/signup`
- POST `/api/auth/signin`

routes/*auth.routes.js*

```
const { verifySignUp } = require("../middleware");
const controller = require("../controllers/auth.controller");

module.exports = function(app) {
  app.use(function(req, res, next) {
    res.header(
      "Access-Control-Allow-Headers",
      "x-access-token, Origin, Content-Type, Accept"
    );
    next();
  });

  app.post(
    "/api/auth/signup",
    [
      verifySignUp.checkDuplicateUsernameOrEmail,
      verifySignUp.checkRolesExisted
    ],
    controller.signup
  );

  app.post("/api/auth/signin", controller.signin);
};
```

Authorization:

- GET `/api/test/all`
- GET `/api/test/user` for loggedin users (user/moderator/admin)
- GET `/api/test/mod` for moderator
- GET `/api/test/admin` for admin

routes/*user.routes.js*

```
const { authJwt } = require("../middleware");
const controller = require("../controllers/user.controller");

module.exports = function(app) {
  app.use(function(req, res, next) {
    res.header(
      "Access-Control-Allow-Headers",
      "x-access-token, Origin, Content-Type, Accept"
    );
    next();
  });

  app.get("/api/test/all", controller.allAccess);

  app.get(
    "/api/test/user",
    [authJwt.verifyToken],
    controller.userBoard
  );

  app.get(
    "/api/test/mod",
    [authJwt.verifyToken, authJwt.isModerator],
    controller.moderatorBoard
  );

  app.get(
    "/api/test/admin",
    [authJwt.verifyToken, authJwt.isAdmin],
    controller.adminBoard
  );
};
```

Don't forget to add these routes in *server.js*:

```
...
// routes
require('./app/routes/auth.routes')(app);
require('./app/routes/user.routes')(app);

// set port, listen for requests
...
```

Run & Test with Results

Run Node.js application with command: `node server.js`

Tables that we define in *models* package will be automatically generated in MySQL Database.

If you check the database, you can see things like this:

```
mysql> describe users;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| id        | int(11)      | NO   | PRI | NULL    | auto_increment |
| username  | varchar(255) | YES  |     | NULL    |                |
| email     | varchar(255) | YES  |     | NULL    |                |
| password  | varchar(255) | YES  |     | NULL    |                |
| createdAt | datetime     | NO   |     | NULL    |                |
| updatedAt | datetime     | NO   |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+


mysql> describe roles;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| id        | int(11)      | NO   | PRI | NULL    |       |
| name      | varchar(255) | YES  |     | NULL    |       |
| createdAt | datetime     | NO   |     | NULL    |       |
| updatedAt | datetime     | NO   |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+


mysql> describe user_roles;
+-----------+----------+------+-----+---------+-------+
| Field     | Type     | Null | Key | Default | Extra |
+-----------+----------+------+-----+---------+-------+
| createdAt | datetime | NO   |     | NULL    |       |
| updatedAt | datetime | NO   |     | NULL    |       |
| roleId    | int(11)  | NO   | PRI | NULL    |       |
| userId    | int(11)  | NO   | PRI | NULL    |       |
+-----------+----------+------+-----+---------+-------+
```

If you don't use `initial()` function in Sequelize `sync()` method. You need to run following SQL script:

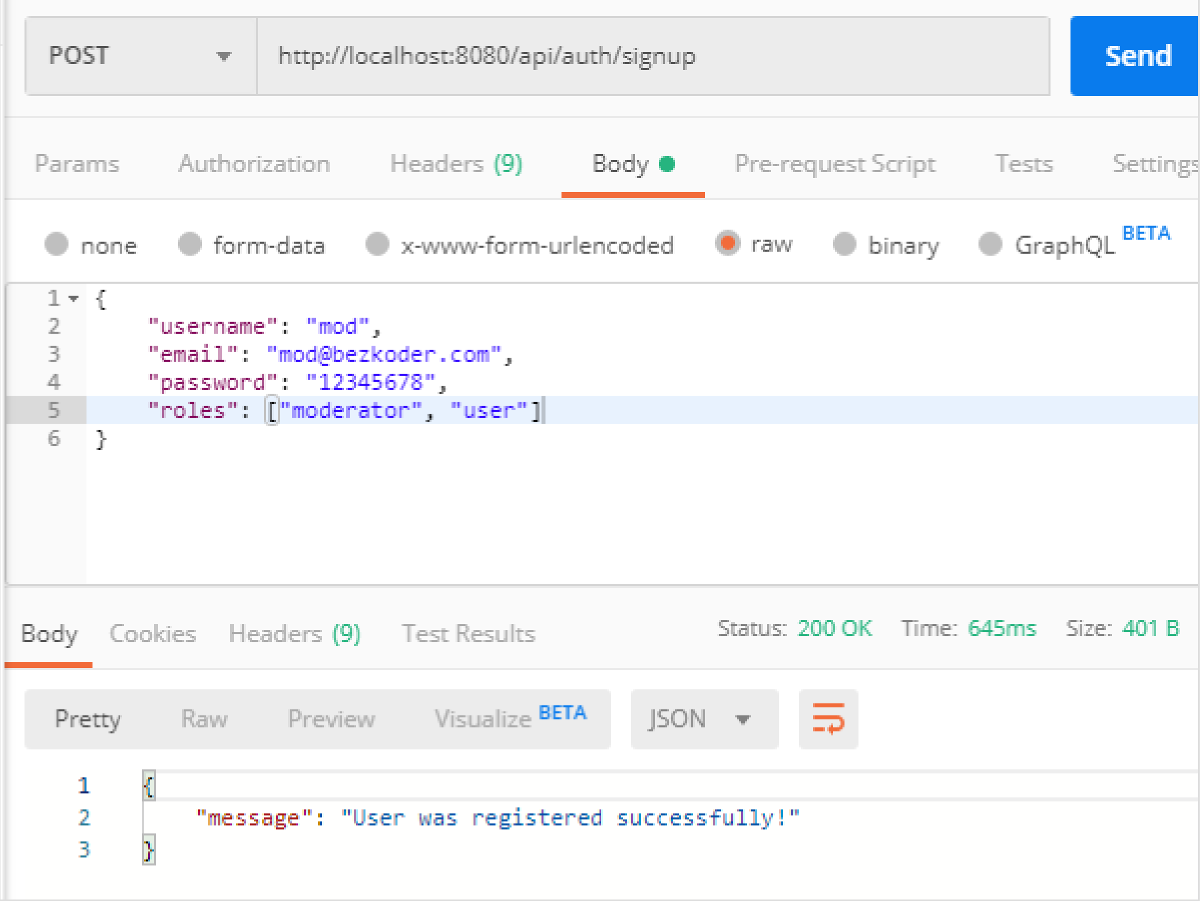
```
mysql> INSERT INTO roles VALUES (1, 'user', now(), now());
mysql> INSERT INTO roles VALUES (2, 'moderator', now(), now());
mysql> INSERT INTO roles VALUES (3, 'admin', now(), now());
```

3 records will be created in `roles` table:

```
mysql> select * from roles;
+----+-----------+---------------------+---------------------+
| id | name      | createdAt           | updatedAt           |
+----+-----------+---------------------+---------------------+
|  1 | user      | 2020-01-13 09:05:39 | 2020-01-13 09:05:39 |
|  2 | moderator | 2020-01-13 09:05:39 | 2020-01-13 09:05:39 |
|  3 | admin     | 2020-01-13 09:05:39 | 2020-01-13 09:05:39 |
+----+-----------+---------------------+---------------------+
```

Register some users with `/signup` API:

- **admin** with `admin` role
- **mod** with `moderator` and `user` roles
- **zkoder** with `user` role



http://localhost:8080/api/auth/signup

```
{
    "username": "mod",
    "email": "mod@bezkoder.com",
    "password": "12345678",
    "roles": ["moderator", "user"]
}
```

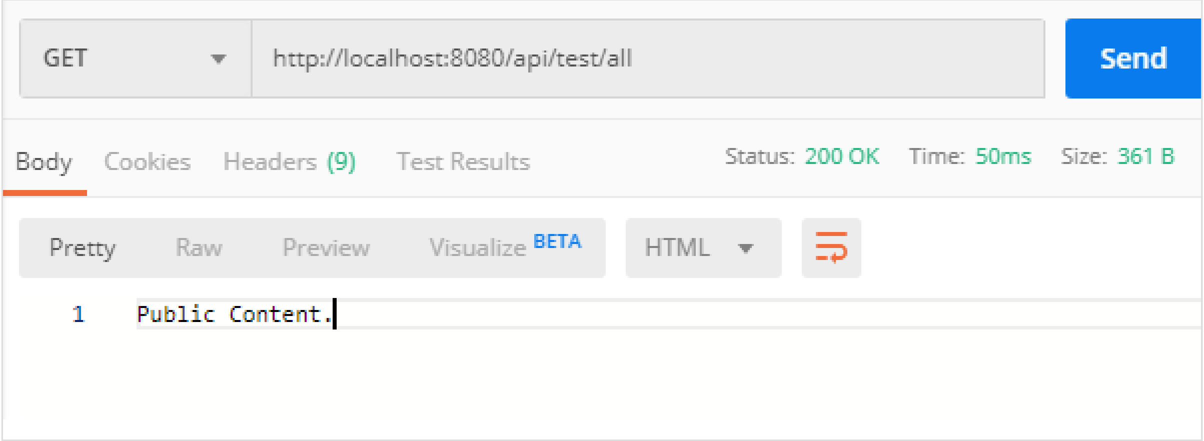
Status: 200 OK Time: 645ms Size: 401 B

```
{
    "message": "User was registered successfully!"
}
```

Our tables after signup could look like this.

```
mysql> select * from users;
+----+----------+--------------------+---------------------------------------------------
| id | username | email              | password
+----+----------+--------------------+---------------------------------------------------
|  1 | admin    | admin@bezkoder.com | $2a$08$w3cYCF.N0UQZO19z8CQSZ.whzxFS5vMoi9k51g3TQx9
|  2 | mod      | mod@bezkoder.com   | $2a$08$tTj1l28esAxPSSvl3YqKl./nz35vQF7Y76jGtzcYUhH
|  3 | zkoder   | user@bezkoder.com  | $2a$08$U2F07dLyYZjzTxQbFMCAcOd1k8V1o9f6E4TGVJHpy0V
+----+----------+--------------------+---------------------------------------------------


mysql> select * from user_roles;
+---------------------+---------------------+--------+--------+
| createdAt           | updatedAt           | roleId | userId |
+---------------------+---------------------+--------+--------+
| 2020-01-13 09:22:01 | 2020-01-13 09:22:01 |      1 |      2 |
| 2020-01-13 09:23:25 | 2020-01-13 09:23:25 |      1 |      3 |
| 2020-01-13 09:22:01 | 2020-01-13 09:22:01 |      2 |      2 |
| 2020-01-13 09:21:51 | 2020-01-13 09:21:51 |      3 |      1 |
+---------------------+---------------------+--------+--------+
```

Access public resource: GET `/api/test/all`

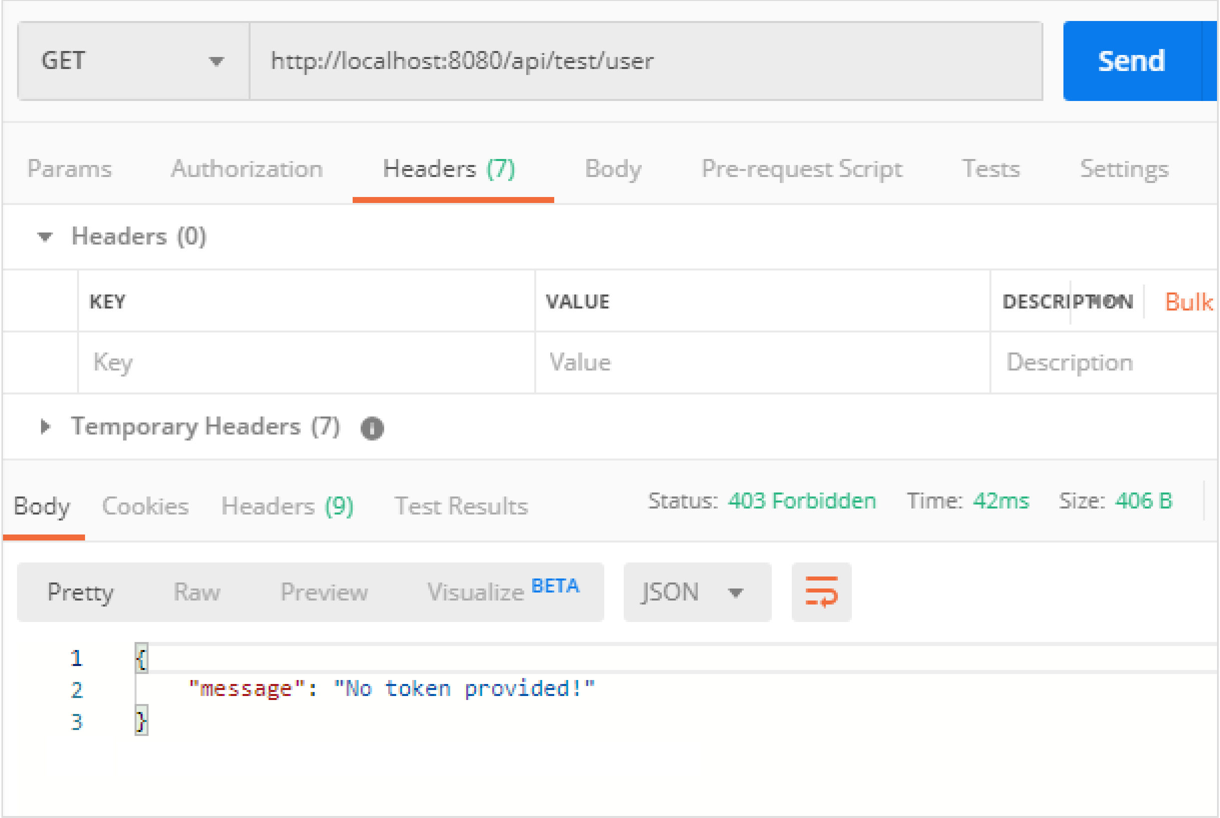


http://localhost:8080/api/test/all

Status: 200 OK Time: 50ms Size: 361 B

```
Public Content.
```

Access protected resource: GET `/api/test/user`

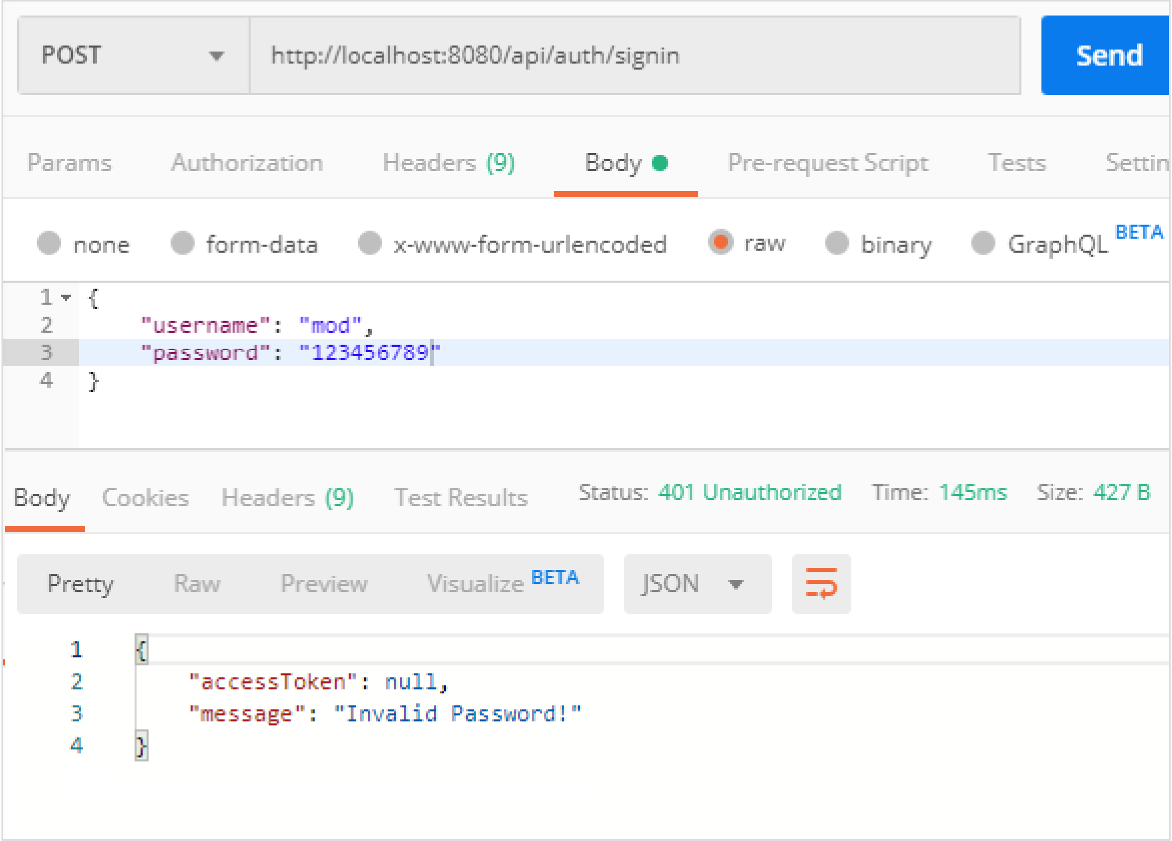


GET http://localhost:8080/api/test/user Send

Status: 403 Forbidden Time: 42ms Size: 406 B

```
{
    "message": "No token provided!"
}
```

Login an account (with wrong password): POST `/api/auth/signin`



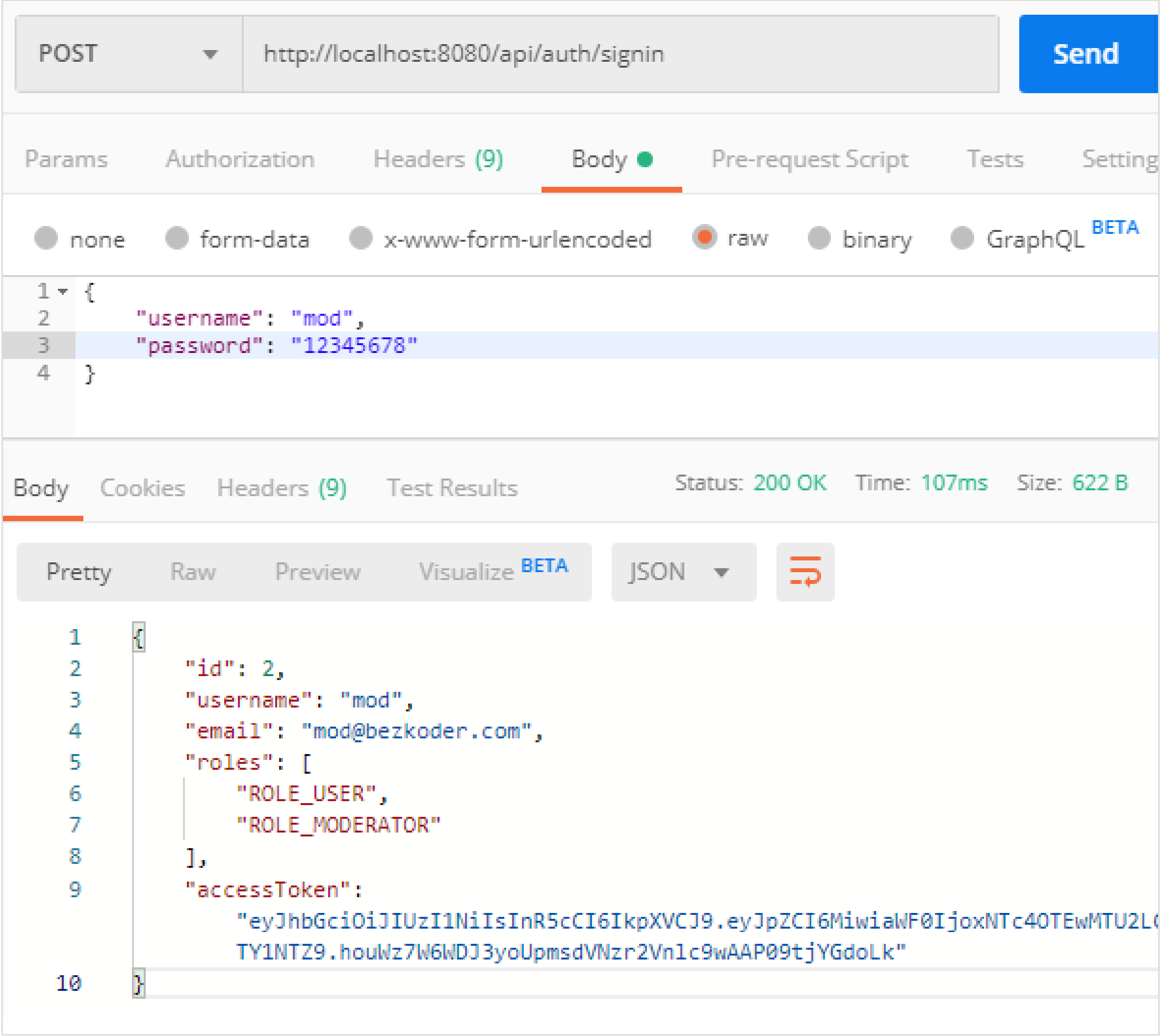
POST http://localhost:8080/api/auth/signin Send

```
{
    "username": "mod",
    "password": "123456789"
}
```

Status: 401 Unauthorized Time: 145ms Size: 427 B

```
{
    "accessToken": null,
    "message": "Invalid Password!"
}
```

Login an account: POST `/api/auth/signin`



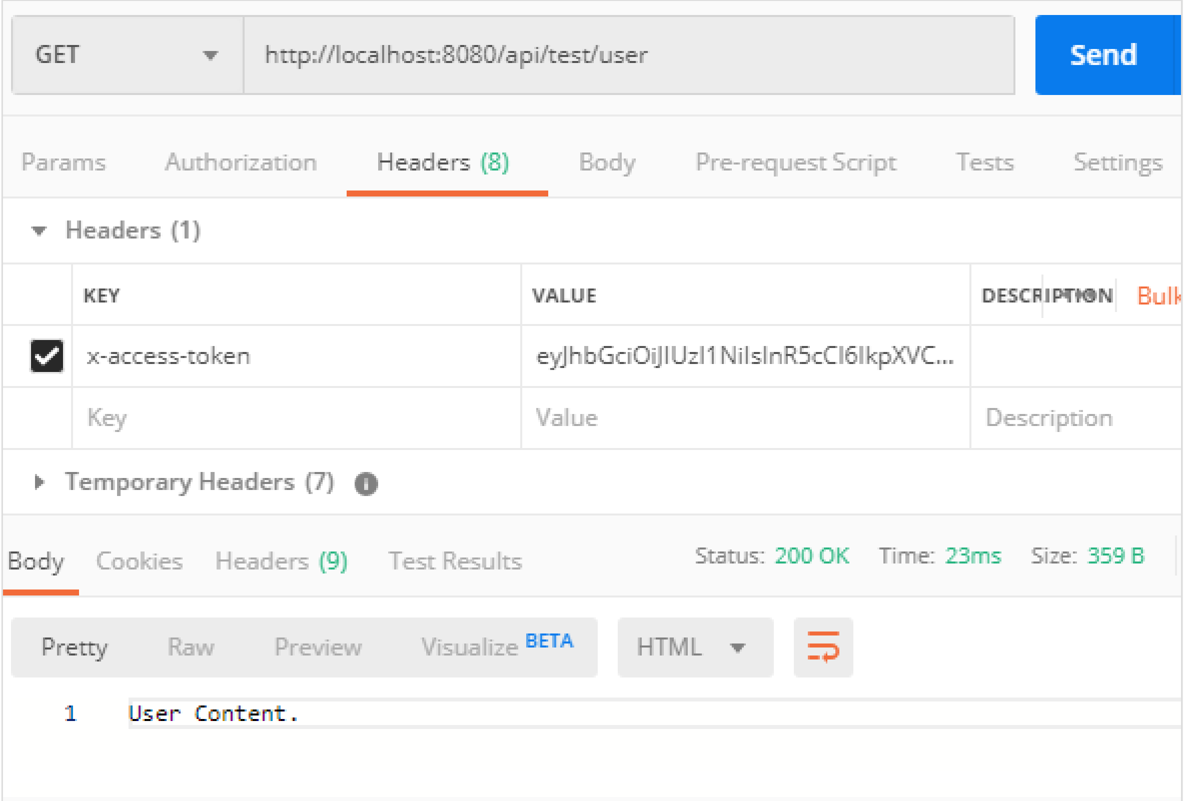
POST http://localhost:8080/api/auth/signin

```
{
    "username": "mod",
    "password": "12345678"
}
```

Status: 200 OK Time: 107ms Size: 622 B

```
{
    "id": 2,
    "username": "mod",
    "email": "mod@bezkoder.com",
    "roles": [
        "ROLE_USER",
        "ROLE_MODERATOR"
    ],
    "accessToken":
        "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MiwiaWF0IjoxNTc4OTEwMTU2L
        TY1NTZ9.houWz7W6WDJ3yoUpmsdVNzr2Vnlc9wAAP09tjYGdoLk"
}
```

Access protected resources: GET `/api/test/user`



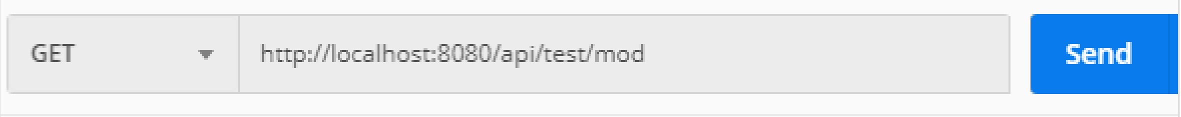
GET http://localhost:8080/api/test/user

Headers (1)

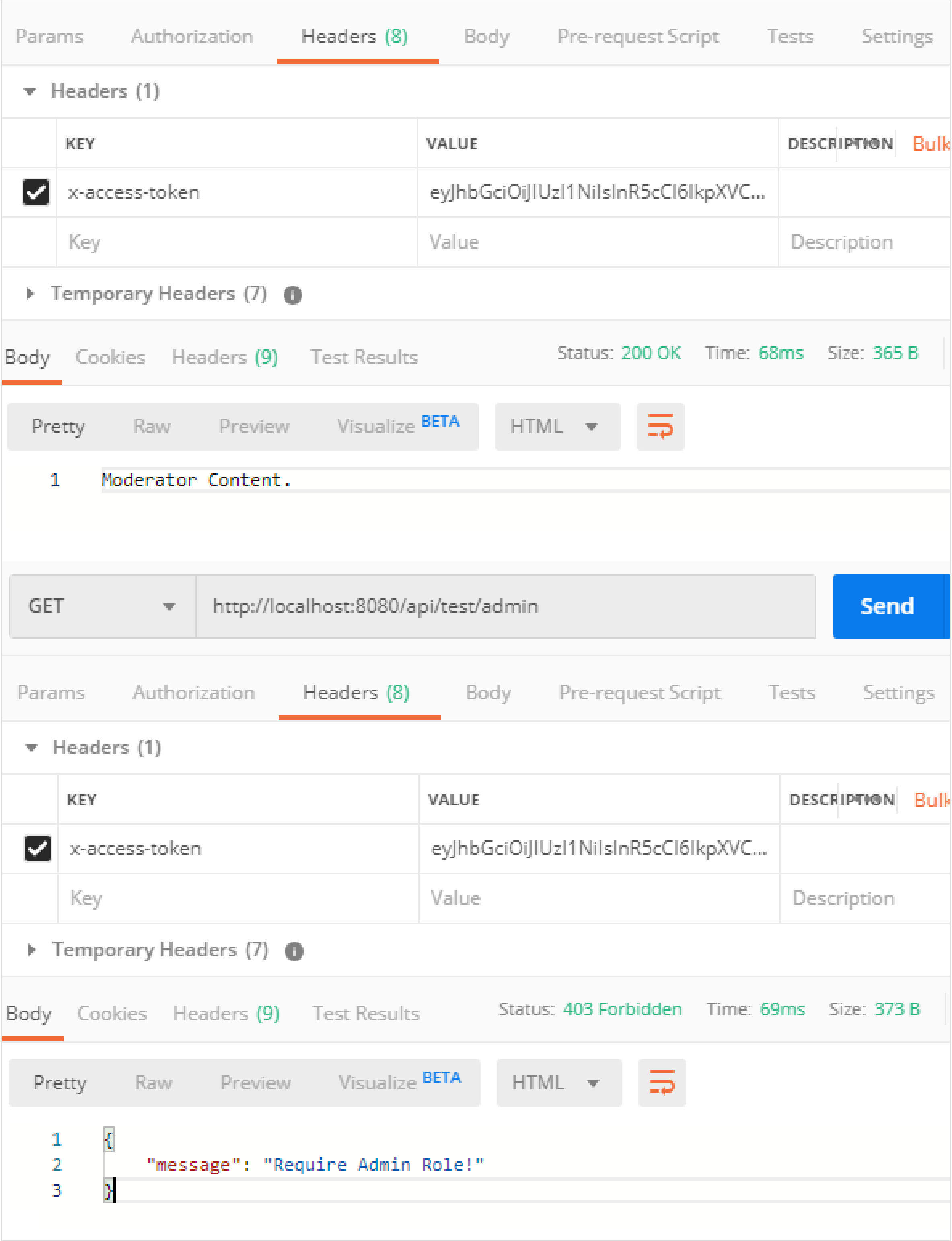
	KEY	VALUE	DESCRIPTION
☑	x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC...	
	Key	Value	Description

Status: 200 OK Time: 23ms Size: 359 B

```
User Content.
```



GET http://localhost:8080/api/test/mod



	KEY	VALUE	DESCRIPTION
☑	x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC...	
	Key	Value	Description

Status: 200 OK Time: 68ms Size: 365 B

```
Moderator Content.
```

GET http://localhost:8080/api/test/admin

	KEY	VALUE	DESCRIPTION
☑	x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC...	
	Key	Value	Description

Status: 403 Forbidden Time: 69ms Size: 373 B

```
{
    "message": "Require Admin Role!"
}
```

Conclusion

Congratulation!

Today we've learned so many interesting things about Node.js Token Based Authentication with JWT - JSONWebToken in just a Node.js Express Rest Api example.
Despite we wrote a lot of code, I hope you will understand the overall architecture of the application, and apply it in your project at ease.

You should continue to know how to implement Refresh Token:
JWT Refresh Token implementation in Node.js example (https://bezkoder.com/jwt-refresh-token-node-js/)

If you need a working front-end for this back-end, you can find Client App in the post:
- Vue.js JWT Authentication with Vuex and Vue Router (https://bezkoder.com/jwt-vue-vuex-authentication/)
- Angular 8 JWT Authentication example with Web Api (https://bezkoder.com/angular-jwt-authentication/)
- Angular 10 JWT Authentication example with Web Api (https://bezkoder.com/angular-10-jwt-auth/)
- Angular 11 JWT Authentication example with Web Api (https://bezkoder.com/angular-11-jwt-auth/)
- Angular 12 JWT Authentication example with Web Api (https://www.bezkoder.com/angular-12-jwt-auth/)
- React JWT Authentication (without Redux) example (https://bezkoder.com/react-jwt-auth/)
- React Hooks: JWT Authentication (without Redux) example (https://bezkoder.com/react-hooks-jwt-auth/)
- React Redux: JWT Authentication example (https://bezkoder.com/react-redux-jwt-auth/)

Happy learning! See you again.

Further Reading

- https://www.npmjs.com/package/express (https://www.npmjs.com/package/express)
- http://expressjs.com/en/guide/routing.html (https://expressjs.com/en/guide/routing.html)
- In-depth Introduction to JWT-JSON Web Token (https://bezkoder.com/jwt-json-web-token/)
- Sequelize Associations (https://sequelize.org/master/manual/associations.html)

Fullstack CRUD Application:
- 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://www.bezkoder.com/angular-12-node-js-express-mysql/)
- React + Node.js + Express + MySQL example (https://bezkoder.com/react-node-express-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/)

Source Code

You can find the complete source code for this tutorial on Github (https://github.com/bezkoder/node-js-jwt-auth).

authentication (https://www.bezkoder.com/tag/authentication/)

authorization (https://www.bezkoder.com/tag/authorization/)

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

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

registration (https://www.bezkoder.com/tag/registration/) rest api (https://www.bezkoder.com/tag/rest-api/)

security (https://www.bezkoder.com/tag/security/) sequelize (https://www.bezkoder.com/tag/sequelize/)

token based authentication (https://www.bezkoder.com/tag/token-based-authentication/)

167 thoughts to “Node.js Express: JWT example | Token Based Authentication & Authorization”

Encrypter

January 30, 2020 at 9:33 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-274)

Great job, thanks a lot!
Just one question, can we integrate this with sequelize migrate due to the fact that it makes the job easier for entitites

REPLY

bezkoder

January 31, 2020 at 8:30 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-278)

Yes, we can. But in this tutorial, I want to keep everything simple and clean, so we don’t use sequelize migrate 🙂

REPLY

Encrypter

February 4, 2020 at 9:26 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-292)

Thanks a lot, this is the best explanation I’ve found, I’m learning a lot from your articles, If I will find you on social media (Medium and others) I’ll be following you.

REPLY

Hirwa Blessing

September 26, 2020 at 2:05 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4860)

Great work, thanks a bunch.
I have a simple question, you’re using those methods which you didn’t define like user.getRoles() but you never defined these methods. I tried to read on sequelize

custom setters and getters but about setters and getters they must be defined on a model but these were used without definition so I wanted to know where they're coming from.

REPLY

M. Tayyab Ferozi

November 27, 2020 at 4:43 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5575)

This is the magic of sequelize, please read the docs

REPLY

Pedro Schmitt

February 13, 2020 at 4:03 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-354)

Great walkthrough! Helped a lot.
I like the way that you organize the models, controllers and routes in every tutorial.

REPLY

bezkoder

February 13, 2020 at 11:50 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-355)

Thanks! I'm so happy to here that.

REPLY

Ben

March 5, 2020 at 6:14 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-462)

Great Tutorial, thank you very much! It helped very much with my project 🙂 I still have one question though – I hope you'll be able to help:

Everything is working and I am trying to build a page where an admin can change the User Roles of a specific user. Right now, I have no Idea what to change or how I would implement this on the backend site. How can I update a User and send a new array of roles to the backend and update the belongsToMany association?

Thanks in advance!

REPLY

bezkoder
March 6, 2020 at 2:04 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-464)

Hi, you can add a new route with `[authJwt.verifyToken, authJwt.isAdmin]` middlewares. This route calls controller method that updates just one table: *user_roles*.

REPLY

Encrypter
May 10, 2020 at 3:10 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2204)

Hi Ben, I'm trying to implement the same thing.
Did you achieve a solution to it?

REPLY

Encrypter
May 11, 2020 at 8:31 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2226)

Solved, thanks a lot Bezkoder!

REPLY

Patrick Seewald
March 6, 2020 at 9:45 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-465)

Hi, really Great Tutorial, but i have the problem, that the Signup doesnt work

I checked out your source code from github, but i get the following message:

Unhandled rejection Error: WHERE parameter "username" has invalid "undefined" value

I think it is inside the verifysignup middleware – maybe you can imagine whats wrong?

Thanks in Advance

REPLY

bezkoder
March 6, 2020 at 2:00 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-467)

Hi, how did you send POST request? Did you set appropriate Header: `Content-Type: application/json` ?

REPLY

Shannon Lee
January 5, 2021 at 6:40 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-6090)

Thank you! In postman, I had to change the content type from Text to JSON, and now it works 🙂

For those looking for it, it's below the text field where you input "http://localhost:8080/api/auth/signup"

REPLY

Labeeb Farooq
March 3, 2021 at 7:03 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-7358)

Hello
Do we need to use POSTMAN to put this link
http://localhost:8080/api/auth/signup (http://localhost:8080/api/auth/signup) ?

REPLY

Rahul Sahni
May 18, 2020 at 12:35 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2569)

shift these lines:
// routes
require('./app/routes/auth.routes')(app);
require('./app/routes/user.routes')(app);

below :

// 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 }));
```

It will work, the issue is routes are called before parsing JSON so it gets undefined in the body.
Hope this will help someone facing the same issue.
Thanks a lot, bezkoder for this great tutorial 🙂

REPLY

Robson
May 20, 2020 at 2:33 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2691)

Did not work.

TypeError: Cannot read property 'username' of undefined

REPLY

Arman
September 23, 2020 at 10:03 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4825)

Try to restart node server..

REPLY

Luky
June 22, 2020 at 6:57 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3397)

Thankyou Rahul, your solution works just awesome

REPLY

tony
March 12, 2020 at 11:41 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-506)

Hi. I have a problem whereby every time I start the server through "node server.js" command everything in the database gets reset. It's making development a bit harder. Can you help with this??

REPLY

bezkoder
March 12, 2020 at 1:17 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-507)

Hi, you can change:

```
db.sequelize.sync({force: true})
```

to:

```
db.sequelize.sync()
```

REPLY

Mrudang
March 13, 2020 at 4:02 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-513)

I am not able to signup and login based on User Mod,Admin

REPLY

bezkoder
March 14, 2020 at 8:51 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-528)

Hi, please make sure that you've inserted 3 rows into Roles table first.

REPLY

Labeeb Farooq
March 8, 2021 at 7:05 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-7476)

hi, their are three rows in roles table with user, moderator and admin names but still when i signup on application, user always becomes user not admin or moderator?
Please help me with this problem?

REPLY

Louis Philippe Facun
March 16, 2020 at 2:11 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-546)

Can we have a version without serialize please? I already made my tables using serialize messes my erd design 🙁

REPLY

bezkoder

March 18, 2020 at 1:01 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-562)

Hi, you can find it here: Build Node.js Rest APIs with Express & MySQL (https://bezkoder.com/node-js-rest-api-express-mysql/)

REPLY

Khairul

March 23, 2020 at 10:17 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-600)

Hi, really Great Tutorial, but i have the problem,
when Access protected resources: GET /api/test/mod
Access protected resources: GET /api/test/mod
Error: Unhandled rejection TypeError: Cannot read property 'getRoles' of null

REPLY

Khairul

March 23, 2020 at 10:18 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-601)

Access protected resources: GET /api/test/admin

Error: Unhandled rejection TypeError: Cannot read property 'getRoles' of null

REPLY

bezkoder

March 23, 2020 at 11:35 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-602)

Hi, did you run `initial()` function to create 3 row in `roles` table?

REPLY

cherif
March 25, 2020 at 1:24 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-635)

HI, I love your work i found this walk through complete.

REPLY

Lomas
April 3, 2020 at 4:51 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-721)

Thank you so much, this is one of the best Node.js JWT tutorial I've read!

REPLY

lightfan
April 4, 2020 at 2:12 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-728)

Thanks for sharing your code for this Node.js JWT auth. It works like a charm!

REPLY

orion
April 11, 2020 at 7:39 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-835)

Great article loved it..how would implement it with refresh tokens? Thanks in advance

REPLY

> **bezkoder**
> August 8, 2021 at 11:19 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-10550)
>
> Hi, here you are: JWT Refresh Token implementation in Node.js example (https://www.bezkoder.com/jwt-refresh-token-node-js/)
>
> REPLY

Sunflower
April 12, 2020 at 9:52 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-841)

Bezkoder, your tutorials are just awesome, thank you very much for your work 🙂
Maybe u have this version without roles(admin, moder) only user?
If yes, could you please share it?

REPLY

bezkoder
April 14, 2020 at 3:36 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-856)

Hi, admin or moderator is just a role. You can remove them in the define steps of the model.
Try your best to do it. 🙂

REPLY

MOHSEN
April 17, 2020 at 5:59 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-891)

Thanks for this useful tutorial. When I run the node server.js after cloned the source code from GitHub I have following error. Unfortunately I couldn’t find any solution for that. Do you have any suggestions for that?

```
Unhandled rejection SequelizeConnectionRefusedError: connect ECONNREFUSED 127.0.0.1:3306
at
C:\Users\ShojaMo\webApp\tasktracker\node_modules\sequelize\lib\dialects\mysql\connection-manager.js:123:19
at tryCatcher
(C:\Users\ShojaMo\webApp\tasktracker\node_modules\bluebird\js\release\util.js:16:23)
at Promise._settlePromiseFromHandler
(C:\Users\ShojaMo\webApp\tasktracker\node_modules\bluebird\js\release\promise.js:547:31)
at Promise._settlePromise
(C:\Users\ShojaMo\webApp\tasktracker\node_modules\bluebird\js\release\promise.js:604:18)
at Promise._settlePromise0
(C:\Users\ShojaMo\webApp\tasktracker\node_modules\bluebird\js\release\promise.js:649:10)
at Promise._settlePromises
(C:\Users\ShojaMo\webApp\tasktracker\node_modules\bluebird\js\release\promise.js:725:18)
at _drainQueueStep
```

```
(C:\Users\ShojaMo\webApp\tasktracker\node_modules\bluebird\js\release\async.js:93:12)
at _drainQueue
(C:\Users\ShojaMo\webApp\tasktracker\node_modules\bluebird\js\release\async.js:86:9)
at Async._drainQueues
(C:\Users\ShojaMo\webApp\tasktracker\node_modules\bluebird\js\release\async.js:102:5)
at Immediate.Async.drainQueues [as _onImmediate]
(C:\Users\ShojaMo\webApp\tasktracker\node_modules\bluebird\js\release\async.js:15:14)
at processImmediate (internal/timers.js:456:21)
```

REPLY

bezkoder

April 17, 2020 at 6:02 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-892)

Hi, please make sure that you've already run MySQL database with correct configuration.

REPLY

MOHSEN

April 17, 2020 at 6:28 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-894)

Thank you for quick response 🙂

REPLY

Felipe

April 30, 2020 at 9:25 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1847)

Hi, I'm connecting in my db normaly using MySQL Workbench, but when I try to conect the application with database I have the follow error:

Unhandled rejection SequelizeConnectionRefusedError: connect ECONNREFUSED 127.0.0.1:3306

How said Mohsen , infortunately I couldn't find any solution for that. Do you have any others suggestions for that?

REPLY

Felipe
April 30, 2020 at 9:38 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1848)

Resolved! My config db was wrong kkkkkkk

REPLY

MOHSEN
April 17, 2020 at 6:17 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-893)

Actually, I ran the code on Win and Mac and both have the same issue! It looks like something is wrong when it tries to run the database. Thanks in advance for any suggestions.

REPLY

Patrickson
April 21, 2020 at 1:06 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1015)

Hi Mr Bezkoder,
Congratulations on this excellent tutorial. That’s what I’ve been looking for ever since.
However, I have a hard time integrating this Backend into uen Development FrontEnd React with Redux and JWT Passport.

Could you help me? I’m new to ReactJS Development.
Thank you

REPLY

bezkoder
April 21, 2020 at 3:15 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1034)

Hi, I will write the tutorial when having time 🙂

REPLY

Megha
April 22, 2020 at 8:16 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1123)

how can i integrate in your existing crud application model? that is tutorials?

REPLY

bezkoder
April 22, 2020 at 10:51 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1283)

Hi, you can create more routes and controllers, then apply auth middleware on the routes 🙂

REPLY

Kusuma
April 29, 2020 at 6:24 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1793)

Hi really great tutorial.
how to insert data to multiple tables at a time – parent/child relationship in nodejs to mySql ?
Could you help me? I'm new to node.js Development.

Relationship between Product & ProductDetails tables.
Sample JSON object:

```
{
"productName":"laptop",
"inv_date":"2020-04-01",
"productDetails":[
        {
            "name":"HP 15.6inc 64GB",
            "price":34052
        },
        {
            "name":"Dell 16.4inc 64GB",
            "price":37761
        },
        {
            "name":"Acer 14inc Spin 3",
            "price":32464
        },
        {
            "name":"Refurb HP Spectre x360 ",
            "price":54150
        }
    ]
}
```

REPLY

bezkoder
April 29, 2020 at 11:56 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1802)

Hi, I think you should rename the entities first, for example:
– ProductName -> Category
– ProductDetails -> Product

Then you can read this tutorial for working with One-to-Many Relationship in Node.js:
Sequelize One-to-Many Association example with Node.js & MySQL (https://bezkoder.com/sequelize-associate-one-to-many/)

REPLY

Kusuma
May 1, 2020 at 10:28 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1855)

Thank you so much , its improves productivity in less time and works fine 🙂

REPLY

Kusuma
May 2, 2020 at 7:27 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-1870)

Hi,
Really it’s a good tutorial.Is there anyway to generate(swagger integration) some default documentation without writing the API Documentations (comments) manually? Like in ASP.NET core.

Sample swagger documentation comments in each routing page:
/**
* @swagger
* /registerUser:
* post:
* tags:
* – Users
* name: Register
* summary: Register a new user
* consumes:
* – application/json

* produces:
* – application/json
* parameters:
* – name: body
* in: body
* schema:
* $ref: '#/definitions/User'
* type: object
* properties:
* first_name:
* type: string
* last_name:
* type: string
* username:
* type: string
* email:
* type: string
* password:
* type: string
* format: password
* required:
* – username
* – email
* – password
* responses:
* '200':
* description: User created
* '403':
* description: Username or email already taken
*/

Thanks in advance.

REPLY

Venkatesh Krishnappa
July 18, 2020 at 4:45 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3805)

You can check out Sequelize, an ORM for Node. It is easy to use and provides ability to represent such relationships

REPLY

Manish Upadhyaya
May 8, 2020 at 4:40 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2086)

I am having an issue with `user.setRoles()` is not a function in my `signUp` request:

{
“message”: “user.setRoles is not a function”
}

REPLY

bezkoder
May 8, 2020 at 6:34 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2092)

Hi, please make sure that you’ve initialized Sequelize object and models correctly.

REPLY

TJ
May 22, 2020 at 4:31 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2753)

Hi i have the same error, I was wondering if you could explain where the setRoles function is coming from, is it automatically generated?

REPLY

bezkoder
May 24, 2020 at 1:43 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2868)

Hi, it is automatically generated by Sequelize 🙂

REPLY

SofaDev39
June 8, 2020 at 6:47 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3152)

Hi, thanks for your tutorial 🙂
I fixed this issue with an alias :

```
db.user.belongsToMany(db.role, {
  through: "Table_User_Roles",
  foreignKey: "idUser",
  otherKey: "idRole",
  as: "roles"
});
```

I hope that help you 😉

Milquebits
May 16, 2020 at 1:05 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2494)

Love your tutorials! Quick question, can you explain why the corsOptions.origin is set to http://localhost:8081 (http://localhost:8081), when the server is setup to listen on http://localhost:8080 (http://localhost:8080)?

You stated: “Notice that we set origin: http://localhost:8081 (http://localhost:8081)“, but did not provide an explanation of why this is different from the listening port.

REPLY

bezkoder
May 19, 2020 at 3:28 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2604)

If you use a REST Client such as Postman, you call API from the same origin `http://localhost:8080` . It is ok and CORS doesn’t need to work here.

But when you send HTTP request from another origin (another front-end app with another port: Angular, React, Vue…), for example: `http://localhost:8081` , you must accept the Cross-origin resource sharing (CORS). It is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served.

REPLY

anis
May 17, 2020 at 10:10 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2515)

Hi,

I am very new to backend. I follow the tutorial and also clone the repo but always showing the message “Coluld not get any response”. Please let me know what should I have to do. do I need to install any software along. like mySql?

REPLY

bezkoder
May 19, 2020 at 3:24 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2603)

Hi, the example works with MySQL database, so you MUST install MySQL and configure appropriate db parameters 🙂

REPLY

james swafford
May 18, 2020 at 2:48 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2576)

bezkoder, excellent tutorial. Complete and well thought out. I’m facing an issue that you have appeared to solve, but for the life of me, I cant replicate from your tutorial into my own code.

I have react/redux front-end with express.js api.
Login works and returns a token.
Attempts to access endpoints with the x-access-token fail from localhost, but succeed from PostMan.

Attempts from localhost give the below error:
Access to fetch at ‘https:///dev/buyer’ from origin ‘http://localhost:3000’ has been blocked by CORS policy: Request header field x-access-token is not allowed by Access-Control-Allow-Headers in preflight response.

I have the below cors configuration

```
let corsOptions = {
  origin: "http://localhost:3000",
};

app.use(cors(corsOptions));
```

I dont understand why PostMan would work, but my localhost will not.

REPLY

bezkoder

May 19, 2020 at 3:22 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2602)

Hi, your server should return that it accepts custom headers:

```
res.header(
  "Access-Control-Allow-Headers",
  "x-access-token, Origin, Content-Type, Accept"
);
```

REPLY

akhil

May 19, 2020 at 8:34 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2622)

Great tutorial, thanks a lot.
Can you implement this with refresh token rotation method.
Thanks in Advance

REPLY

bezkoder

May 19, 2020 at 9:51 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2628)

Hi, I will write the tutorial when having time 🙂

REPLY

bezkoder

August 8, 2021 at 11:19 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-10549)

Hi, here you are: JWT Refresh Token implementation in Node.js example (https://www.bezkoder.com/jwt-refresh-token-node-js/)

REPLY

Hassaan

May 19, 2020 at 6:47 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2651)

node run server stuck at:
Executing (default): CREATE TABLE IF NOT EXISTS `roles` (`role_id` INTEGER , `role_code` VARCHAR(255), `role_descr` VARCHAR(255), `createdAt` DATETIME NOT NULL, `updatedAt` DATETIME NOT NULL, PRIMARY KEY (`role_id`)) ENGINE=InnoDB;
Executing (default): SHOW INDEX FROM `roles`

REPLY

bezkoder
May 19, 2020 at 10:57 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2664)

Hi, please send more log details about the issue.

REPLY

Hassaan
May 20, 2020 at 6:59 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2675)

Thanks for your quick reply but I've figured out the issue. server was running fine there was some other error in my code.

REPLY

indresh
July 29, 2020 at 8:33 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4065)

i also stuck at ' SHOW INDEX FROM `user_roles`' what is solution for that?

REPLY

Reynald Daffa
May 20, 2020 at 1:18 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2667)

Hi, great tutorial. btw have u tutorial for logout jwt? and is it good to save jwt in the database?

REPLY

bezkoder
May 21, 2020 at 4:19 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2713)

Hi, we're gonna save the Refresh Token in the database. I will write a tutorial for this when having time 🙂

REPLY

AngieA
May 20, 2020 at 9:52 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2702)

Hi bezkoder! Been working with some of your tutorials recently, which have been so great, and I'm trying to get a website up on my namecheap hosting server. When I have the API backend loaded and started server.js, I'm able to successfully register and login through the cPanel terminal using curl commands.

So today I went through your "Vue.js JWT Authentication with Vuex and Vue Router" tutorial, and everything works in testing when connecting to a MySQL database on my local computer, but after loading my build to the server, I'm getting "net::ERR_CONNECTION_REFUSED". At first I was getting a CORS error, so I changed the corsOptions origin value to my domain, thinking that would work. I'm no longer getting the CORS error message.. just "net::ERR_CONNECTION_REFUSED".

Any ideas? Thank you!!

REPLY

bezkoder
May 21, 2020 at 4:17 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2712)

Hi, please show me your browser console log.

REPLY

AngieA
May 21, 2020 at 5:54 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2734)

Thanks for your help!!

https://auel.dev/dinner-robot/ERR_CONNECTION_REFUSED.png (https://auel.dev/dinner-robot/ERR_CONNECTION_REFUSED.png)

REPLY

bezkoder

May 22, 2020 at 7:29 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2742)

Hi, the message said that: “username is already in use!”. So did you try to signup with another one?

REPLY

Rakesh

May 22, 2020 at 11:02 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2745)

Is this a micro service architecture ?

REPLY

krishnan

May 27, 2020 at 9:54 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-2967)

can you write refresh token code

REPLY

bezkoder

August 8, 2021 at 11:25 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-10554)

Hi, here you are: JWT Refresh Token implementation in Node.js example (https://www.bezkoder.com/jwt-refresh-token-node-js/)

REPLY

Spookee

May 30, 2020 at 8:13 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3038)

Thank you very much! Very interesting and useful!!

REPLY

Rayner

June 4, 2020 at 2:27 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3107)

Thanks for the great tutorial! I got this working with SQL Server using tedious and express4-tedious. Didn't need mysql2 so uninstalled it. So I have a route set up in express4-tedious that I would like to only be available if a user (ROLE_USER) is logged in:

```
app.use('/alignments', require('./app/routes/alignments'));
```
(in server.js)

I have nothing against myql. Just need to use a mssql database because of some spatial functions that are not available in mysql, for example:

```
/* GET alignments listing. */
router.get('/', function (req, res) {
req.sql("select id, fid, GeomCol1.AsTextZM() as alignmentwkt, GeomCol1.STSrid as srid, aligngrp1, aligngrp2, alignment, GeomCol1.STLength() as length, lrs.STStartMeasure(GeomCol1) as startm, lrs.STEndMeasure(GeomCol1) as endm, lrs.STMeasureRange(GeomCol1) as measure_range, hascurve, numsegs, thumbnail, mapurl from alignmentstest4 order by aligngrp1 asc, aligngrp2 asc, alignment asc for json path")
.into(res, '[]');
});
```
(in routes/alignments.js)

So how would you go about making this new /alignments route available to only users who are logged in?

Going further, there will be some POSTs and PUTs that I would like only moderators or admins to be able to get access to:

```
/* PUT update station/offset/point. */
router.put('/ptso/put/update/:ptid', textParser, function (req, res) {
req.sql("exec update_pt_so_h @ptid, @pts")
.param('ptid', req.params.ptid, TYPES.Int)
.param('pts', req.body, TYPES.NVarChar)
.exec(res);
});
```

Any pointers you can give will be appreciated. I'm very new to Express and have been working with React for a couple of months.

Thanks again for a great tutorial!

REPLY

Rayner

June 8, 2020 at 3:29 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3147)

How would you add another protected route without sequelize?

I have a route /alignments that I would like to add as a protected route. I'm trying to get it to work like the user route. I tried to add it to /user/alignments but that didn't work either. Please help!

REPLY

Mark

June 9, 2020 at 3:59 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3164)

If i see well in this app everybody can register as admin.

REPLY

Bobby

June 11, 2020 at 1:08 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3215)

Hi, first of all, thank you for your guidance. Can you make a tutorial about Node.js Token Based Authentication & Authorization example without ORM Sequelize? Thank you!

REPLY

hargon

June 11, 2020 at 2:06 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3226)

This article is great.

The checkDuplicateUserNameOrEmail function check email only when username is already in use. Maybe you should check email only when username is not in use.

Thanks.

REPLY

sady

June 16, 2020 at 4:05 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3295)

Great work, quick question, why when i add a new model (table), it still enforces createdAt and updatedAt fields to be there? how can i remove this enforcement pls? much thankkkks

REPLY

rmg
November 24, 2020 at 8:00 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5538)

createdAt: {
type: Sequelize.DATE,
field: 'creado'
},

REPLY

Rayner
June 20, 2020 at 3:09 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3376)

I got this to work on my local development machine using SQL Server (MSSQL) rather than MySQL. Also using Tedious and Express4/Tedious. Now trying to publish the Express app to Azure and unable to get to any of the 'auth' routes. It returns Cannot GET /api/auth/signin and Cannot GET /api/auth/signup. It does let me have access to the /api/test/user and /api/test/mod and /api/test/admin (these show {"message":"No token provided!"}) and /api/test/all (shows Public Content.) All my other routes are protected, so they also show up as {"message":"No token provided!"}. How does one go about getting the auth routes to work in Azure?

REPLY

Krzysiu
June 22, 2020 at 10:04 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3398)

First i want thenks for tutorial – it work great … but i dont have rules. I think i dont create or add something. I have profile component from github like everything else. When i try find ROLES_USER, ROLES_ADMIN or ROLES_MODERATOR i cant find it in my code. If i dont give what i have in my code for you please tell me and i add in next comment. Do you know what i miss ?Sorry i dont add "my" code but i dont know what i should add and want know exacly what create ROLES.

PS
Sorry for my english – i know im terrible 😛

REPLY

Steffan
June 29, 2020 at 10:16 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3527)

Thanks a lot for the Tutorial, it helped me understand a lot I didn't grasp before. It would be nice if you could make this post a bit clearer:
React + Node.js Express: User Authentication with JWT example
Scrolling through half the page was a little bit confusing to find the implementation.
Besides that I have one question maybe you or someone else here can answer.

```
const authJwt = {
verifyToken: verifyToken,
isAdmin: isAdmin,
isModerator: isModerator,
isModeratorOrAdmin: isModeratorOrAdmin
};
```

Everything in these is undefined for me and thus the arrow functions also don't work. Do you have any idea why that could be the case? I think I'm probably missing something super simple but I don't know what.

Thanks again for the great tutorial, even with my fails it was easy for me to follow and made some things clear.

REPLY

vr14
June 30, 2020 at 5:40 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3549)

Hi greatesolution i tried the CRUD operations and it worked perfectly . I have an error in the api while trying the authentication with mysql. I'm not able to add a new user in both mysql db as well as in server (i.e) in the locahost:8080/api/auth/signup in server . in my db i'm getting the following error
Unhandled rejection SequelizeForeignKeyConstraintError: Cannot add or update a child row: a foreign key constraint fails (`mysql`.`user_roles`, CONSTRAINT `user_roles_ibfk_1` FOREIGN KEY (`roleId`) REFERENCES `roles` (`id`) ON DELETE CASCADE ON UPDATE CASCADE)
Help needed. Thanks in advance!!!

REPLY

Michael Layne
March 4, 2021 at 2:27 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-7392)

I came across this myself when doing another one of these excellent tutorials. It may be because you already have data in the table. Try truncating that table (or any related tables as well – after backing up or exporting any data) and then run it and it may work. HTH.

REPLY

Cris
July 3, 2020 at 6:25 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3598)

amazing, thank you very much greetings from Chile!

REPLY

Diogo
July 3, 2020 at 10:51 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3605)

That’s awesome, thanks!

One quick note: You’ve created the folder as “middlewares” but it’s importing it later as “middleware”. 🙂

REPLY

Harish
July 4, 2020 at 10:34 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3611)

Hi,
I’m getting an error, When I start the server node server.js. Please find the below error log:

```
D:\Personal\NodeJS\node-js-jwt-auth>node server.js
D:\Personal\NodeJS\node-js-jwt-auth\server.js:23
db.sequelize.sync();
^
TypeError: Cannot read property ‘sync’ of undefined
at Object. (D:\Personal\NodeJS\node-js-jwt-auth\server.js:23:14)
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)
at internal/main/run_main_module.js:17:47
Thanks.
Harish

REPLY

Harish
July 4, 2020 at 10:45 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3612)

Hi,
I don’t know how to do this initialization.
If anyone has source code, Kindly the share to me.
Thank you so much for these tutorials.

Thanks,
Harish

REPLY

Chris
July 4, 2020 at 10:12 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3617)

Everything looks great except I fear many people using this may not realize that adding roles in the signup route is a bad idea. Anyone could hit the api of the site using postgres and create an account with admin privileges if they knew the website used this. I would highly recommend removing removing everything in this if statement and only keeping what is in the else (default User role) in the signup function in auth.user.js

```
if (req.body.roles) {
        Role.findAll({
          where: {
            name: {
              [Op.or]: req.body.roles
            }
          }
        }).then(roles => {
          user.setRoles(roles).then(() => {
            res.send({ message: "User was registered successfully!" });
          });
        });
      } else {
        // user role = 1
        user.setRoles([1]).then(() => {
          res.send({ message: "User was registered successfully!" });
        });
      }
```

REPLY

bezkoder

July 5, 2020 at 7:38 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3623)

Hi, this is just for development and understand how to run authorization. In real app, we don’t accept anybody signup for all roles.

REPLY

Malcolm

August 8, 2020 at 3:41 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4241)

Hi,

For my use case it’s ideal as I expect the admin to do the signups and allocate roles, so the signup route will be admin role protected.

REPLY

Aak

July 9, 2020 at 7:22 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3661)

Do you have similar example using just mysql and not sequilize? Also can the above examples done using mysql?

REPLY

Malcolm
August 8, 2020 at 2:58 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4240)

Hi Aak,

I've just completed this tutorial by converting to using node-postgres rather than use sequelize, so it will be possible. As I'm new to most of this myself and haven't used MySql I can't be specific but it should just be a matter of creating the appropriate SQL statements for each of the database queries required.

Best regards,

REPLY

Coder
July 11, 2020 at 4:30 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3695)

hi, my token is throwing a 500 internal server error. It's only the token that's supposed to go in yeah?

REPLY

Coder
July 11, 2020 at 5:38 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3697)

fixed it, sorry. Thank you so much for this

REPLY

Alex
July 19, 2020 at 7:39 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3826)

Hi Bezkoder,

Thanks for this tutorial, very helpful. But I don't understand one of the functions.
What does 'getRoles()' from authJwt.js file do?
I cannot see where it is defined or referenced in the project.

REPLY

bezkoder
July 19, 2020 at 9:54 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3828)

Hi, you can see that we use `belongsToMany()` so that Sequelize supports the methods `getRoles()` for model User.

REPLY

Alex
July 20, 2020 at 12:37 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-3842)

Thank you. I just found more info in the docs: https://sequelize.org/master/class/lib/associations/belongs-to-many.js~BelongsToMany.html (https://sequelize.org/master/class/lib/associations/belongs-to-many.js~BelongsToMany.html), after your comment.
I am trying my best to deconstruct your project without sequelize because I already have an ERD diagram prepared with views, procedures, triggers. I feel this would be more efficient for the DB.

REPLY

Malcolm
August 8, 2020 at 2:51 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4239)

Hi and thanks bezkoder for this tutorial, it has helped me understand this process better considering I've rewritten parts to use node-postgres rather than sequelize since I'm more familiar with SQL than working with ORMs. The conversion "forced" me to learn more Postgresql functionality. Now to tackle the React frontend part where I'll be using material-ui and formik but still following your React Hooks: JWT Authentication (without Redux) example. Excellent work.

REPLY

santosh
August 12, 2020 at 9:14 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4281)

Hey thanks @bezKoder for such good blog

REPLY

obi (http://none)
August 26, 2020 at 4:21 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4466)

please i need an explanation; whats the setRole() for, is it a sequelize function.

```
.then(roles => {
user.setRoles(roles).then(() => {
res.send({ message: “User was registered successfully!” });
});
});
```

REPLY

carlo veneziano
September 2, 2020 at 9:34 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4586)

Thank you for this article! you made my day <3

REPLY

Sebastian
September 4, 2020 at 2:32 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4604)

Hi copied your repository and ran it but I got no entries on roles table

REPLY

bezkoder
September 5, 2020 at 2:41 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4606)

Hi, you should change the code in *server.js* to:

```
// db.sequelize.sync();
// force: true will drop the table if it already exists
db.sequelize.sync({force: true}).then(() => {
  console.log('Drop and Resync Database with { force: true }');
  initial();
});
```

It will create 3 necessary rows first. 🙂

REPLY

Sebastian

September 6, 2020 at 2:12 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4610)

thanks, great tutorial 🙂

REPLY

khoi tran

September 10, 2020 at 7:08 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4671)

UnhandledPromiseRejectionWarning: SequelizeDatabaseError: Cannot drop table 'roles' referenced by a foreign key constraint 'user_roles_ibfk_1' on table 'user_roles'. . this is happen when i try to execute server.js . I know this is an error when i try to delete table that has foreign key , how to fix this in node js .

REPLY

bezkoder

September 10, 2020 at 3:29 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4674)

Hi, try to drop `user_roles` table first.

REPLY

Bojan

September 17, 2020 at 12:37 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4767)

How you get roles in middlewares/authJwt.js, since you don't include const Role?

REPLY

bezkoder
September 18, 2020 at 12:24 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4772)

Hi, please look at how we define Sequelize Associations in `app/models/index.js`. The method `user.getRoles()` is supported without our implementation.

REPLY

James
September 22, 2020 at 12:53 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4810)

Great guide thank you!
Posting this again as you may have missed it.

In your explanation of folder stucture, you have "middleware".
However you refer to this folder throughout the guide as "middlewares" (plural), when we are creating the files authJwt, index, and verifySignUp.

But, inside the code, you are importing files from the folder "middleware" (singular).

Would advise changing to avoid any confusion on why code isn't working if people are not checking this.

Again, love all the guides and options you are providing for different stacks.
Best,

REPLY

bezkoder
September 22, 2020 at 8:27 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4813)

Hi, thank you for you comment. I've just fixed that typo 🙂

REPLY

James

September 24, 2020 at 12:48 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4826)

I also wanted to add to this as I had an issue with the correct board being displayed per the users role.
(User, Moderator, Admin)
In the auth-header.js file you have the following object being returned to the user.service.js GET requests:

return { Authorization: "Bearer " + user.accessToken };

This results in an error of "No token provided". as it is looking for the header "x-access-token", not Authorization.bearer

I have changed it to reflect the following, which resulted in a correct display of the board being viewed by the users role.

return { "x-access-token" : user.accessToken };

If the way in which you have described the object in the guide is correct, could you explain a bit further on how that works?
Because I received the error stated above, I have kept my chgange I made until further notice.

Again, thank you!! 🙂

REPLY

bezkoder

September 24, 2020 at 9:27 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4830)

Hi, it is because I use the frontend for 2 backend:
– Spring Boot: using Bearer token
– Node.js: using x-access-token

I write this notification in the frontend tutorial.

REPLY

Moses

September 24, 2020 at 3:12 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4835)

hey bezkoder,

I have this error :

```
TypeError: Cannot read property ‘username’ of undefined
at exports.signin (C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\app\controllers\auth.controller.js:46:26)
at Layer.handle [as handle_request] (C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\node_modules\express\lib\router\layer.js:95:5)
at next (C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\node_modules\express\lib\router\route.js:137:13)
at Route.dispatch (C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\node_modules\express\lib\router\route.js:112:3)
at Layer.handle [as handle_request] (C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\node_modules\express\lib\router\layer.js:95:5)
at C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\node_modules\express\lib\router\index.js:281:22
at Function.process_params (C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\node_modules\express\lib\router\index.js:335:12)
at next (C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\node_modules\express\lib\router\index.js:275:10)
at C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\app\routes\auth.routes.js:10:5
at Layer.handle [as handle_request] (C:\Users\me\Desktop\env_node\JWT_mysql_node_tut\node_modules\express\lib\router\layer.js:95:5)
```

and unable to solve it till now. Any solution would be appreciated. Thank you!

REPLY

Serena Martin

September 29, 2020 at 4:56 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4884)

This was incredibly useful. I’ve recently started learning NodeJs. It’s an awesome framework.

REPLY

Ahsan

October 3, 2020 at 8:32 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4914)

When i use post api -> localhost:8080/api/auth/signup

```
{
“username” : “ahmad”,
“email” : “ahmad@gmail.com”,
“password”: “1245678”,
“roles” : [“admin”, “user”]
}
```

I face this error

```
Unhandled rejection Error: WHERE parameter “username” has invalid “undefined” value
at MySQLQueryGenerator.whereItemQuery (E:\test\node-js-jwt-auth-master\node_modules\sequelize\lib\dialects\abstract\query-generator.js:2184:13)
at E:\test\node-js-jwt-auth-master\node_modules\sequelize\lib\dialects\abstract\query-generator.js:2173:25
at Array.forEach ()
at MySQLQueryGenerator.whereItemsQuery (E:\test\node-js-jwt-auth-master\node_modules\sequelize\lib\dialects\abstract\query-generator.js:2171:35)
at MySQLQueryGenerator.getWhereConditions (E:\test\node-js-jwt-auth-master\node_modules\sequelize\lib\dialects\abstract\query-generator.js:2583:19)
at MySQLQueryGenerator.selectQuery (E:\test\node-js-jwt-auth-master\node_modules\sequelize\lib\dialects\abstract\query-generator.js:1315:28)
at QueryInterface.select (E:\test\node-js-jwt-auth-master\node_modules\sequelize\lib\query-interface.js:1127:27)
at E:\test\node-js-jwt-auth-master\node_modules\sequelize\lib\model.js:1759:34
at tryCatcher (E:\test\node-js-jwt-auth-master\node_modules\bluebird\js\release\util.js:16:23)
at Promise._settlePromiseFromHandler (E:\test\node-js-jwt-auth-master\node_modules\bluebird\js\release\promise.js:547:31)
at Promise._settlePromise (E:\test\node-js-jwt-auth-master\node_modules\bluebird\js\release\promise.js:604:18)
at Promise._settlePromise0 (E:\test\node-js-jwt-auth-master\node_modules\bluebird\js\release\promise.js:649:10)
at Promise._settlePromises (E:\test\node-js-jwt-auth-master\node_modules\bluebird\js\release\promise.js:729:18)
at _drainQueueStep (E:\test\node-js-jwt-auth-master\node_modules\bluebird\js\release\async.js:93:12)
at _drainQueue (E:\test\node-js-jwt-auth-master\node_modules\bluebird\js\release\async.js:86:9)
at Async._drainQueues (E:\test\node-js-jwt-auth-master\node_modules\bluebird\js\release\async.js:102:5)
at Immediate.Async.drainQueues [as _onImmediate] (E:\test\node-js-jwt-auth-master\node_modules\bluebird\js\release\async.js:15:14)
at processImmediate (internal/timers.js:456:21)
```

REPLY

Ahsan

October 4, 2020 at 3:33 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4917)

Hi!

I have found solution. When I chose raw option then I selected Text instead of Json option.

REPLY

James Lee

October 12, 2020 at 1:44 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5008)

Ok buddy, that’s good

REPLY

Jirka

October 8, 2020 at 8:55 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4958)

Hi and thx for very nice tutorial!
I have problem when I try to login with non-existing user – the node screams about UnhandledPromiseRejectionWarning but I don’t know how and what block should I put inside the try-catch to satisfy node. Do I have to somehow modify app.post(“/api/auth/signin”, controller.signin); ? The findOne already has catch. When I change the exports.signin into exports.signin = async (req, res) => {try {…} catch (err) {..} it isn’t solving the problem. Thank you very much!

```
Executing (default): SELECT “id”, “username”, “email”, “password”, “createdAt”, “updatedAt” FROM “users” AS “users” WHERE “users”.”username” = ‘testuser’ LIMIT 1;
(node:19868) UnhandledPromiseRejectionWarning: Error: WHERE parameter “email” has invalid “undefined” value (…)
at async Function.findOne (node_modules\sequelize\lib\model.js:1917:12)
(node:19868) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an async function without a catch block, or by rejecting a promise which was not handled with .catch(). To terminate the
```

node process on unhandled promise rejection, use the CLI flag `–unhandled-rejections=strict` (see https://nodejs.org/api/cli.html#cli_unhandled_rejections_mode (https://nodejs.org/api/cli.html#cli_unhandled_rejections_mode)). (rejection id: 2) (node:19868) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled will terminate the Node.js process with a non-zero exit code.

REPLY

Snoochie
March 16, 2021 at 8:47 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-7627)

You need Content-Type:application/json in your request header

REPLY

Yugesh Devtare
October 11, 2020 at 7:40 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4993)

Hey, great work!
I had a question, How can we directly set roles for the user using a form from the frontend rather than using postman?

REPLY

Yugesh Devtare
October 11, 2020 at 8:25 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-4998)

How can we directly set roles for the user using a form from the frontend rather than using postman?

REPLY

mave
October 12, 2020 at 9:55 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5007)

Running Ubuntu 20.04

It returns Cannot GET /api/auth/signin and Cannot GET /api/auth/signup. It does let me have access to the /api/test/user and /api/test/mod and /api/test/admin (these show {"message":"No token provided!"}) and /api/test/all (shows Public Content.)

Does anyone know the issue?

Been trying to integrate the https://bezkoder.com/angular-10-jwt-auth/ (https://bezkoder.com/angular-10-jwt-auth/) into this. Does anyone know how to do it? Please share if you have any example?

REPLY

Erick Suryawan
February 9, 2021 at 4:09 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-6911)

I have same problem. try to change usage header in auth-header.js
// return { Authorization: 'Bearer ' + user.accessToken }; // for Spring Boot back-end
return { 'x-access-token': user.accessToken }; // for Node.js Express back-end

REPLY

Lee Soon Fatt
October 24, 2020 at 8:22 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5178)

hi there,

i have an error for "message": "No token provided!" and inside my postman there is no x-access token. is there something that I am missing ? on my postman there is content-type but not x-access

REPLY

Yugesh Devtare
October 29, 2020 at 7:46 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5233)

hello, how can retrieve a list of all the "mod" or "superadmin" from the table?

REPLY

Helmindra
November 1, 2020 at 5:43 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5260)

Hello, I;m stuck in “Unhandled rejection Error: WHERE parameter “username” has invalid “undefined” value”
i do :
// 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 }));

And below it i put :

// routes
require(‘./app/routes/auth.routes’)(app);
require(‘./app/routes/user.routes’)(app);

Trying restart “node server.js” several times like a mention in comment before ,but still no luck still getting that error , maybe someone can help to resolve it ?. – Thanks –

REPLY

DCespedesH
December 6, 2020 at 12:52 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5672)

Hi, as mentioned by Ahsan some days ago, in postman you should change the input from raw text to raw JSON and then you’ll have a success message.

I was stuck with this some time until noticed that.

REPLY

Erick
November 2, 2020 at 3:13 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5266)

Thank you for this tutorial. I’m facing these messages when I try to restart node server.js:
(node:14036) [SEQUELIZE0004] DeprecationWarning: A boolean value was passed to options.operatorsAliases. This is a no-op with v5 and should be removed.

Server is running on port 8080.

(node:14036) UnhandledPromiseRejectionWarning:
SequelizeConnectionRefusedError: connect ECONNREFUSED 127.0.0.1:3306
at ConnectionManager.connect (C:\Users\RARibeiro\OneDrive\ECOSTEEL\node-js-jwt-auth\node_modules\sequelize\lib\dialects\mysql\connection-manager.js:116:17)
at processTicksAndRejections (internal/process/task_queues.js:97:5)

(node:14036) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an async function without a catch block, or by rejecting a promise which was not handled with .catch(). To terminate the node process on unhandled promise rejection, use the CLI flag `–unhandled-rejections=strict` (see https://nodejs.org/api/cli.html#cli_unhandled_rejections_mode (https://nodejs.org/api/cli.html#cli_unhandled_rejections_mode)). (rejection id: 1)

(node:14036) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled will terminate the Node.js process with a non-zero exit code.

REPLY

mark

December 20, 2020 at 8:19 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5856)

I run into the same issue. This shows up when trying to setup the alternative postgres project.

Haven't found a solution yet.

REPLY

bezkoder

December 21, 2020 at 7:10 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5866)

Hi, for Postgres, please visit:
Node.js JWT Authentication with PostgreSQL example (https://bezkoder.com/node-js-jwt-authentication-postgresql/)

REPLY

Subash

December 3, 2020 at 9:11 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5637)

Hello bezkoder, when I try to signup users using Postman, I get a error response saying socket hang up or read Econnreset. Can you help me with it?
All other get requests work perfectly fine.

REPLY

Carlos

December 8, 2020 at 5:31 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5692)

Hello bezkoder … Thanks for this post … Congrats from Colombia!

REPLY

Alan

December 28, 2020 at 1:14 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-5943)

Excellent tutorial. I am trying to convert to typescript, but it has a lot of errors. Do you have any examples with TS?
Thanks,

REPLY

Che

January 12, 2021 at 10:58 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-6210)

Hi Bezcoder,
A very nice tutorials and well documented.

I have issue with ROLE not assigned to a respective (Moderator & admin) user, during signup, is there a suggestion on how to fix that please.

warm regards

REPLY

thanhps

January 14, 2021 at 7:15 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-6238)

Hi, great tutorial. I have one question how to inital roleId & userId ?

REPLY

X

Khamit

March 5, 2021 at 8:49 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-7413)

Hi, Great Tutorial! , can i can add permissions to this project , roles , rules, permissions may be come from database , have y some idea to do it ?

REPLY

Raka

March 18, 2021 at 7:11 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-7654)

Hi,
This such a great tutorial man. So gladly i found it. Btw i wanna ask something, what the next(); meaning in every functions you've make it ?
Thanks.

REPLY

Rappwalk

April 11, 2021 at 5:26 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8041)

Hello, can u help me?, im get stuck here after POST to localhost:3000/api/auth/daftar

```
Executing (default): SELECT `id`, `namalengkap`, `username`, `email`, `password`, `createdAt`, `updatedAt` FROM `penggunas` AS `penggunas` WHERE `penggunas`.`username` = ‘madiajijah7’ LIMIT 1;
Executing (default): SELECT `id`, `namalengkap`, `username`, `email`, `password`, `createdAt`, `updatedAt` FROM `penggunas` AS `penggunas` WHERE `penggunas`.`email` = ‘madiajijah7@gmail.com’ LIMIT 1;
```

there no error in Postman just a seinding request like this :
https://prntscr.com/119eg9i (https://prntscr.com/119eg9i)

REPLY

Rappwalk

April 11, 2021 at 6:02 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8042)

already fix it, forgot to add next() at end the code

REPLY

Terje Rosenlund

May 3, 2021 at 9:14 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8358)

In server.js, line 23:

```
db.sequelize.sync();
// force: true will drop the table if it already exists
// db.sequelize.sync({force: true}).then(() => {
// console.log('Drop and Resync Database with { force: true }');
// initial();
// });
```

The call to initial() is never done so db.roles stays empty causing no insert into db.user_roles when a new user is inserted using /api/auth/signup

REPLY

Subbu Swaroop
May 4, 2021 at 4:57 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8382)

Hey There. A very great article which helped me from the beginning till the end.

Cheers.

REPLY

Remmy
May 7, 2021 at 11:44 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8418)

Hello Bezkoder,
Great job,kudos for your effort.

I am new to MERN with mysql,and Sequelize,I tried to follow the steps in this tutorial,but it seems I have missed some important step,that prevent me from creating a new user,each time I execute the code,I get "TypeError: User.create is not a function" error message at User.create({}) of auth.controller.js. Kindly assist in resolving this issue.

Thank you

REPLY

Leena
May 13, 2021 at 12:56 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8515)

Working great! Thanks alot 🙂

REPLY

Amber
June 1, 2021 at 11:04 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8809)

Hi Bezkoder,

Thank you very much for your tutorial, it is amazing! I followed it step by step and everything worked fine until when I tried to access the protected source. The signup and signin functions work well and I can get the token as well. However, when I tried to do "/api/test/user" (there is a user in my database), it said "no token provided" and when I tried to do"console.log(req.headers["x-access-token"], it was undefined. The same thing happened when I downloaded your source code from git and ran it on my mac. Can you please tell me what happened and how I could fix it?

Thank you so much.

REPLY

lek
June 2, 2021 at 3:52 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8830)

Same
everything working fine but i get the msg: No token provided!

REPLY

Walt
June 13, 2021 at 8:36 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-9052)

Please make sure to select the right back-end (Node or Spring Boot) in the src/services/auth-header.js file of your front-end implementation.

REPLY

lek (http://-)
June 2, 2021 at 2:53 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8827)

Please help: what does it mean?
{//post
"username": "mod",
"email": "mod@test.com",

"password": "test",
"roles": ["moderator", "user"]
}

I get the below message from postman
{
"message": "Conversion failed when converting date and/or time from character string."
}

REPLY

lek
June 2, 2021 at 3:13 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8829)

the datatype or id of roles is not matched the structure then i drop my table and use the initial fuction now is working thank you.

REPLY

lek
June 7, 2021 at 8:21 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8928)

Hi, can anyone help me, i try to select all from my table: i do have tb1.models.js in the models folder

```
exports.test = function (req, res) {
  db.tbl1.findAll({attributes: ['id', 'name']});
};
```

The error here
TypeError: Cannot read property 'findAll' of undefined

REPLY

lek (http://-)
June 7, 2021 at 3:52 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8936)

I found the issue. the code can not connect other tables except: users, user_roles and roles.
in the sql-management the db-account can access all tables.
where is in the code to register other tables?

REPLY

lek (http://-)
June 8, 2021 at 7:23 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8944)

Why the table name system add ‘s’ to the end and add more columns when execute that make me sick and need to findout where is the configuration.
original table name tbl1: id, name

log:
Executing (default): SELECT [id], [name], [createdAt], [updatedAt] FROM [tbl1s] AS [tbl1];

REPLY

lek
June 8, 2021 at 9:06 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-8946)

Finally i found the solution:
add your original table name and set option timestamp in your model.js:

```
name: {
    type: DataTypes.STRING(255),
    allowNull: true
  }
}, {
  tableName: 'tbl1',
  timestamps: false
});
```

NOTED: you can add timestamps: false to the model/index.js as an option to all your models.

REPLY

Joe Vienneau
June 13, 2021 at 11:52 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-9065)

This was very very helpful; thank you!

REPLY

Le Anh Tuan
June 17, 2021 at 7:38 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-9127)

Thanks bezkoder. Very happy with your tutorial

REPLY

amir
July 7, 2021 at 4:10 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-9468)

hi, thanks a lot, I am learning so much!

REPLY

Giuliano
July 19, 2021 at 3:25 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-9737)

Hello bezkoder

I tried the tutorial and I think it is very geat job.

I tried it in my local dev machine (localhost) and then published it on my cloud dev machine.

I wanted to test the tutorial on my dev cloud machine with the usecase I provide old token when signing
so i used the token (for the same user) used some minutest ago in my local dev machine
the token i used before in my local dev machine worked also in my cloud dev machine.

Is it because the token has 24hrs validity ?

could we think to link the token also to the domain of the express application ?

Regards

REPLY

amirmohammad
July 25, 2021 at 8:19 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-9990)

great bezkode i have a question we can control other user or deleat or see profile other user like a admin role ? can u help me or send a tutorial for admin role

REPLY

Kallon

July 25, 2021 at 4:02 pm (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-10010)

Hai bezkoder. You give this tutorial it’s really awesome. I just tried it for the first time and understand. Thanks so much.

REPLY

Mike

July 31, 2021 at 12:39 am (https://www.bezkoder.com/node-js-jwt-authentication-mysql/#comment-10252)

Love your tutorial man. Cannot thank you more for this,

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

‹ Vue Firebase Realtime Database: CRUD example (https://www.bezkoder.com/vue-firebase-realtime-database/)

Vue Firestore: Build a CRUD App example › (https://www.bezkoder.com/vue-firestore-crud/)

Search...

FOLLOW US

(https://facebook.com/bezkoder)

(https://www.youtube.com/channel/UCp0mx9RH0JxaFsmMvKXA86Q)

(https://github.com/bezkoder)

TOOLS

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

X