

Node.js + MongoDB: User Authentication & Authorization with JWT

Last modified: January 8, 2021 (<https://bezkoder.com/node-js-mongodb-auth-jwt/>)  bezkoder (<https://bezkoder.com/author/bezkoder/>)  MongoDB (<https://bezkoder.com/category/mongodb/>), Node.js (<https://bezkoder.com/category/node-js/>), Security (<https://bezkoder.com/category/security/>)

In this tutorial, we're gonna build a Node.js & MongoDB example that supports User Authentication (Registration, Login) & Authorization with JSONWebToken (<https://www.npmjs.com/package/jsonwebtoken>) (JWT). You'll know:

- Appropriate Flow for User Signup & User Login with JWT Authentication
- Node.js Express Architecture with CORS, Authenticaon & Authorization middlewares, Mongoose ODM
- Way to configure Express routes to work with JWT
- How to define Mongoose Models for Authentication and Authorization
- How to use Mongoose to interact with MongoDB Database

Related Posts:

- MERN stack Authentication example (<https://bezkoder.com/react-node-mongodb-auth/>)
- MEAN stack Authentication with Angular 8 example (<https://bezkoder.com/mean-stack-authentication-angular-8/>)
- MEAN stack Authentication with Angular 10 example (<https://bezkoder.com/mean-stack-authentication-angular-10/>)
- MEAN stack Authentication with Angular 11 example (<https://bezkoder.com/mean-stack-authentication-angular-11/>)
- Node.js, Express & MongoDB: Build a CRUD Rest Api example (<https://bezkoder.com/node-express-mongodb-crud-rest-api/>)
- MongoDB One-to-Many Relationship tutorial with Mongoose examples (<https://bezkoder.com/mongoose-one-to-many-relationship/>)
- MongoDB Many-to-Many Relationship with Mongoose examples (<https://bezkoder.com/mongodb-many-to-many-mongoose/>)

Front-end that works well with this:

- Angular 8 (<https://bezkoder.com/angular-jwt-authentication/>)
- Angular 10 (<https://bezkoder.com/angular-10-jwt-auth/>)
- Angular 11 (<https://bezkoder.com/angular-11-jwt-auth/>)
- Vue.js (<https://bezkoder.com/jwt-vue-vuex-authentication/>)
- React (without Redux) (<https://bezkoder.com/react-jwt-auth/>)

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

[Privacy & Policy](#) (<https://bezkoder.com/privacy-policy/>)

[Contents](#) [hide]

Accept

Token Based Authentication

Node.js & MongoDB User Authentication example

Flow for Signup & Login with JWT Authentication

Node.js Express Architecture with Authentication & Authorization

Technology

Project Structure

Create Node.js App

Setup Express web server

Configure MongoDB database

Define the Mongoose Model

Initialize Mongoose

Configure Auth Key

Create Middleware functions

Create Controllers

Controller for Authentication

Controller for testing Authorization

Define Routes

Run & Test with Results

Conclusion

Further Reading

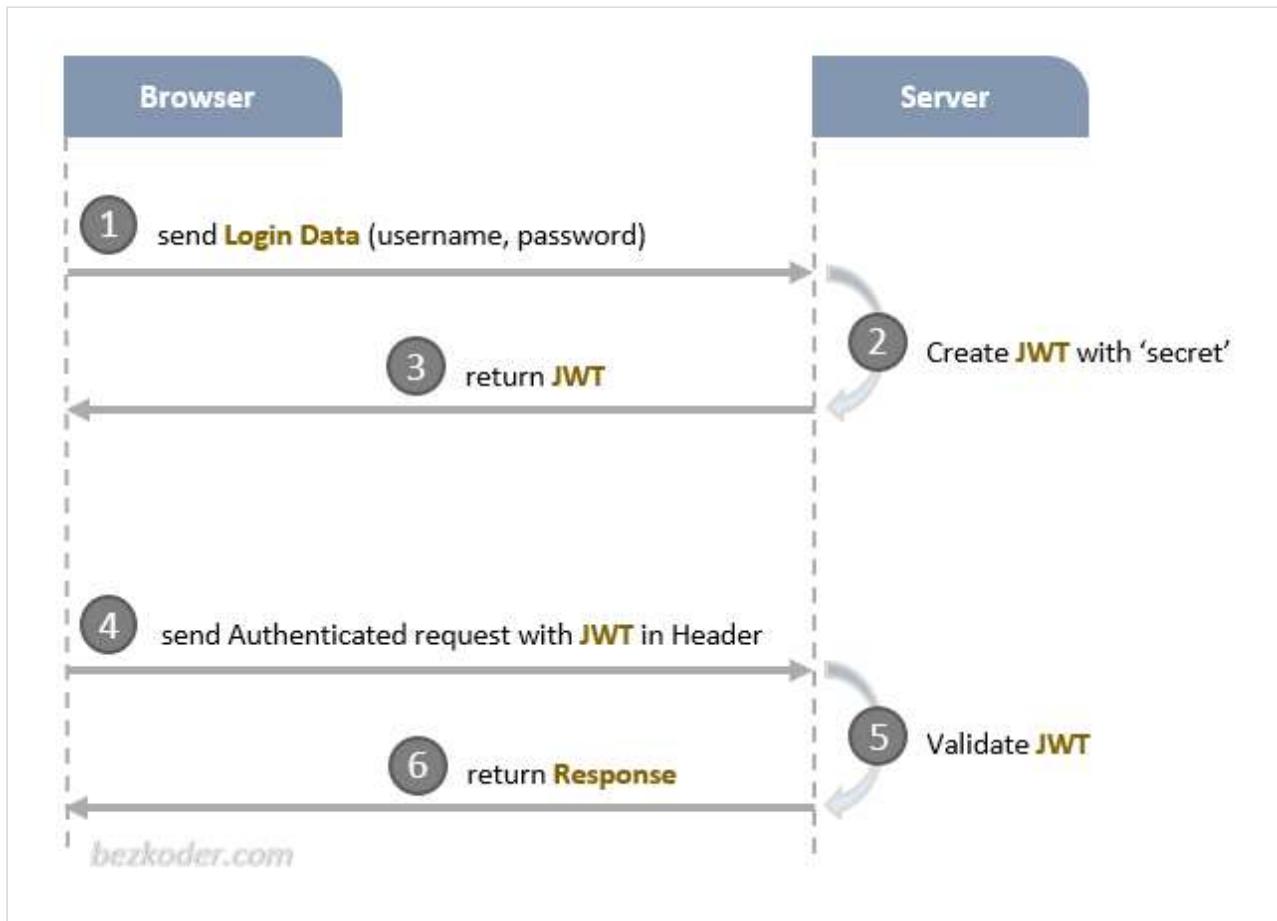
Source Code

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.

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

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



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/>)

Node.js & MongoDB User Authentication example

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

- User can signup new account, or login with username & password.
- By role (admin, moderator, user), the User has access to protected resources or not

These are APIs that we need to provide:

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

Methods

Urls

Actions

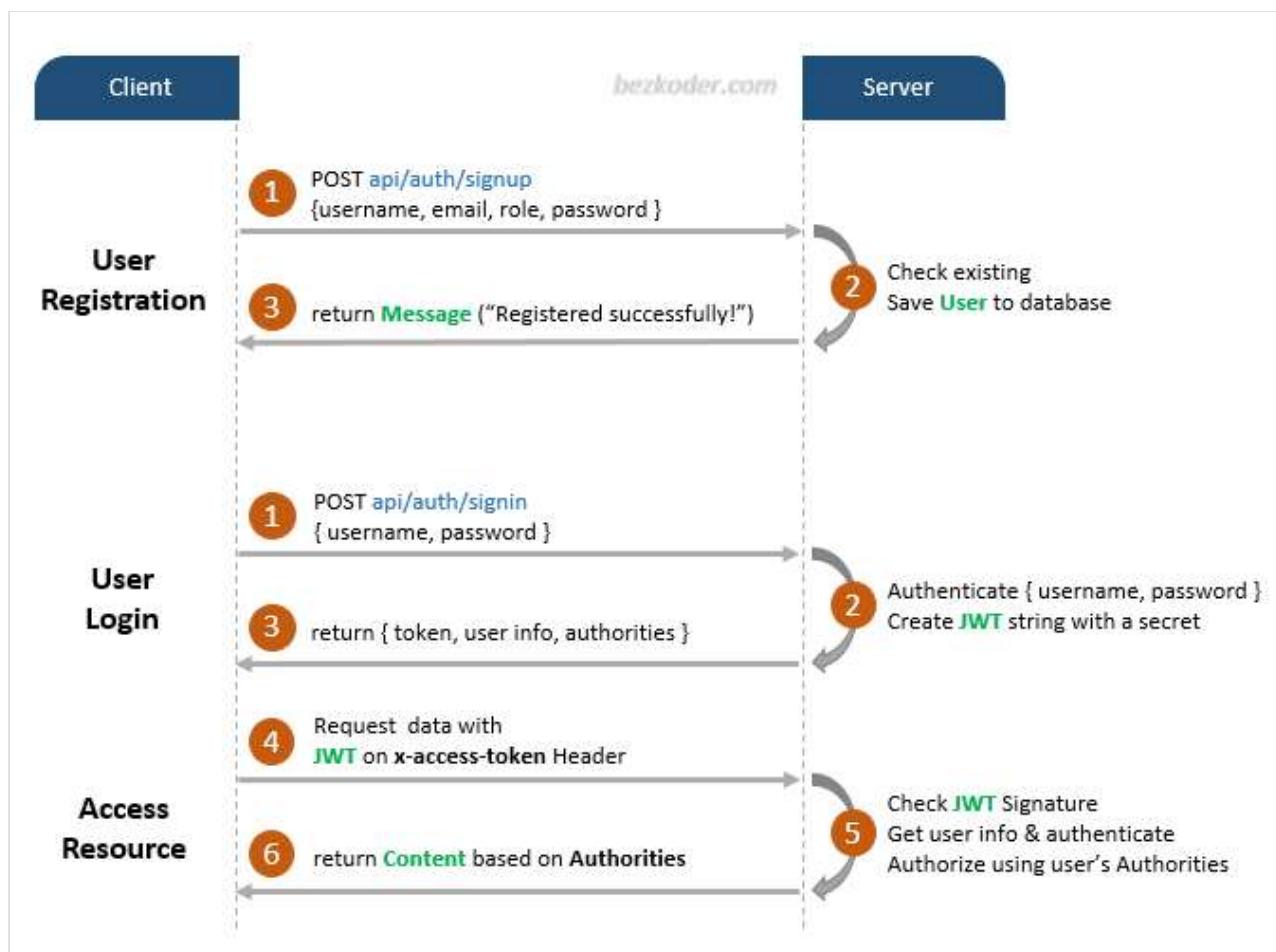
[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

Accept

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

Following diagram shows you the flow that we're gonna implement for User Registration, User Login and Authorization process.



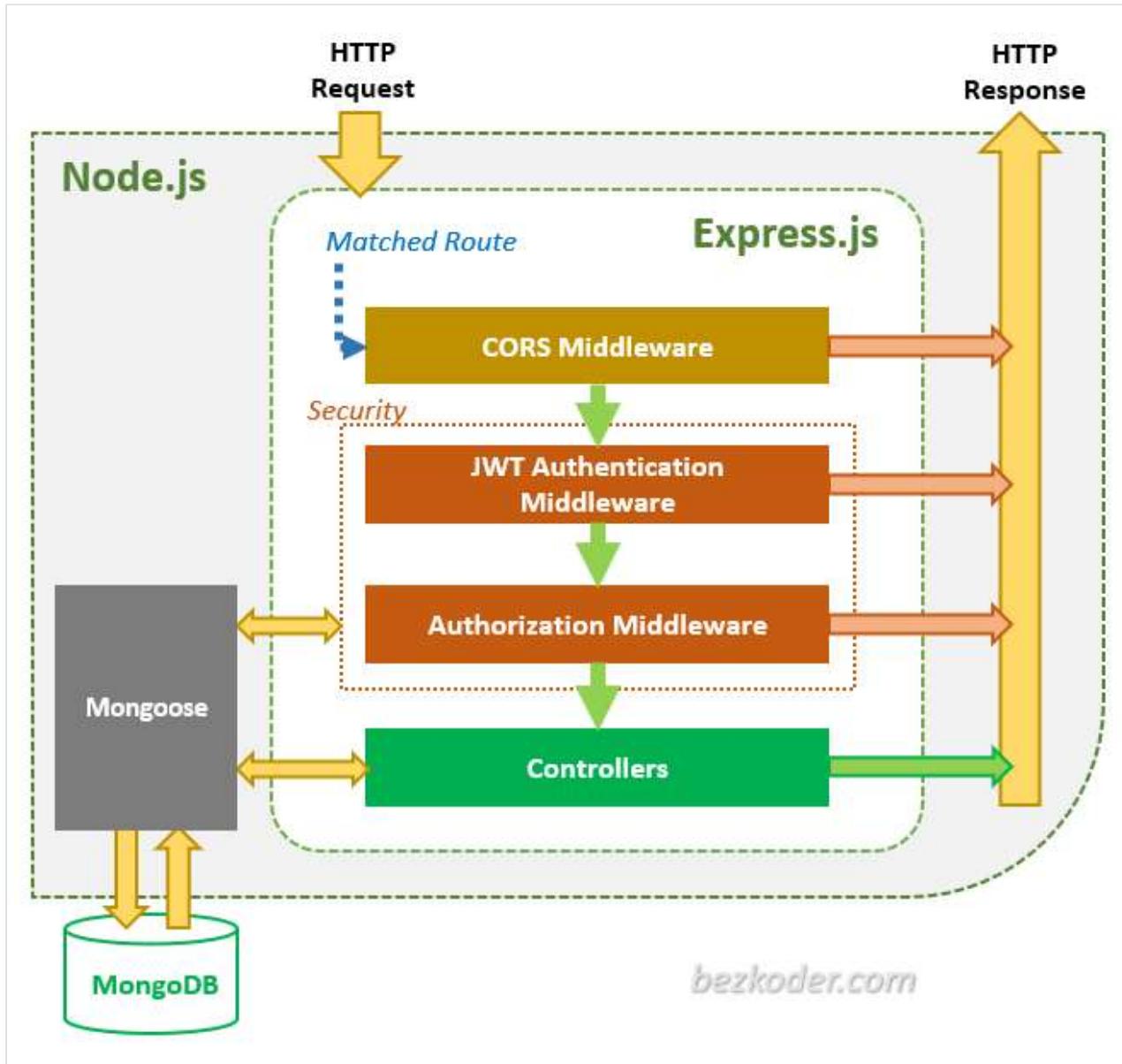
A legal JWT must be added to HTTP **x-access-token** Header if Client accesses protected resources.

Node.js Express Architecture with Authentication & Authorization

We have covered the basic concepts of authentication and authorization. To learn more about Node.js Express, read the full [Node.js Express Documentation](#) (<https://nodejs.org/en/docs/guides/writing-web-applications/>)

Accept

Here is an overview of our Node.js Express App:



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

An error message will be sent as **HTTP response** to Client when the middlewares throw any error, .

Controllers interact with MongoDB Database via **Mongoose** library and send **HTTP response** (token, user information, data based on roles...) to Client.

Technology

We use **Express 4.17.1** to improve your experience with the site. To find out more, you can read the full

- bcryptjs 2.4.3 [Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/) Accept
- jsonwebtoken 8.5.1

- mongoose 5.9.1
- MongoDB

Project Structure

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



Create Node.js App

Create a folder for our project with command:

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

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

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

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

```
npm init

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

Is this ok? (yes) yes
```

Let's install necessary modules such as: `express`, `cors`, `body-parser`, `mongoose`, `jsonwebtoken` and `bcryptjs`.

Run the command:

```
npm install express mongoose body-parser cors jsonwebtoken bcryptjs --save
```

Check `package.json` file, you can see it looks like this:

```
{
  "name": "node-js-jwt-auth-mongodb",
  "version": "1.0.0",
  "description": "Node.js + MongoDB: JWT Authentication & Authorization",
  "main": "server.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [
    "node.js",
    "express",
    "jwt",
    "authentication",
    "mongodb"
  ],
  "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",
    "mongoose": "^5.9.1"
  }
}
```

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

}

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

Accept

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}.`);
});
```

What we've just done in the code above:

- 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

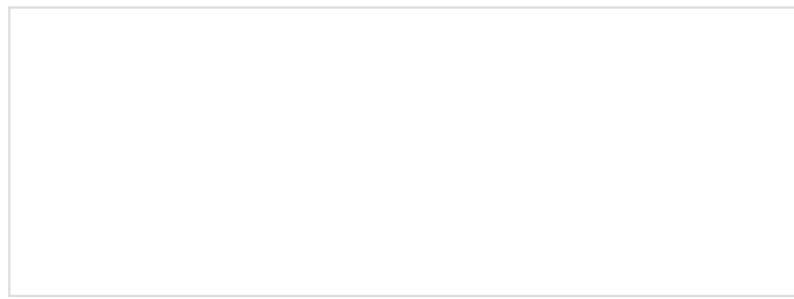
- create an Express app, then add `body-parser` and `cors` middlewares using `app.use()` method. Notice that we set `origin: http://localhost:8081`.

- define a GET route which is simple for test.

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

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:



Configure MongoDB database

In the **app** folder, create **config** folder for configuration.

Then create a new *db.config.js* file that contains parameters for setting up MongoDB later:

```
module.exports = {  
  HOST: "localhost",  
  PORT: 27017,  
  DB: "bezkoder_db"  
};
```

Define the Mongoose Model

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

models/role.model.js

```
const mongoose = require("mongoose");  
  
const Role = mongoose.model(  
  "Role",  
  new mongoose.Schema({  
    name: String  
  })  
);  
  
module.exports = Role;
```

models/user.model.js

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

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

```

const mongoose = require("mongoose");

const User = mongoose.model(
  "User",
  new mongoose.Schema({
    username: String,
    email: String,
    password: String,
    roles: [
      {
        type: mongoose.Schema.Types.ObjectId,
        ref: "Role"
      }
    ]
  })
);

module.exports = User;

```

These Mongoose Models represents **users** & **roles** collections in MongoDB database. User object will have a `roles` array that contains ids in **roles** collection as reference.

This kind is called *Reference Data Models* or *Normalization*. You can find more details at: MongoDB One-to-Many Relationship tutorial with Mongoose examples (<https://bezkoder.com/mongoose-one-to-many-relationship/>)

After initializing Mongoose, we don't need to write CRUD functions because Mongoose supports all of them:

- create a new User: `object.save()`
(https://mongoosejs.com/docs/api/model.html#model_Model-save)
- find a User by id: `User.findById(id)`
(https://mongoosejs.com/docs/api.html#model_Model.findById)
- find User by email: `User.findOne`
(https://mongoosejs.com/docs/api.html#model_Model.findOne)({ email: ... })
- find User by username: `User.findOne({ username: ... })`
- find all Roles which name in given `roles` array: `Role.find`
(https://mongoosejs.com/docs/api.html#model_Model.find)({ name: { \$in: roles } })

These functions will be used in our Controllers and Middlewares.

Initialize Mongoose

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

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

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

```
const mongoose = require('mongoose');
mongoose.Promise = global.Promise;

const db = {};

db.mongoose = mongoose;

db.user = require("./user.model");
db.role = require("./role.model");

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

module.exports = db;
```

Open *server.js* and add following code to open Mongoose connection to MongoDB database:

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

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

```

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

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

db.mongoose
  .connect(`mongodb://${dbConfig.HOST}:${dbConfig.PORT}/${dbConfig.DB}`, {
    useNewUrlParser: true,
    useUnifiedTopology: true
  })
  .then(() => {
    console.log("Successfully connect to MongoDB.");
    initial();
  })
  .catch(err => {
    console.error("Connection error", err);
    process.exit();
  });
}

...
function initial() {
  Role.estimatedDocumentCount((err, count) => {
    if (!err && count === 0) {
      new Role({
        name: "user"
      }).save(err => {
        if (err) {
          console.log("error", err);
        }

        console.log("added 'user' to roles collection");
      });
    }

    new Role({
      name: "moderator"
    }).save(err => {
      if (err) {
        console.log("error", err);
      }

      console.log("added 'moderator' to roles collection");
    });
  });
}

```

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

```

new Role({
  name: "admin"
})Privacy & Policy (https://bezkoder.com/privacy-policy/)
if (err) {

```

```

        console.log("error", err);
    }

    console.log("added 'admin' to roles collection");
});
}
});
}

```

`initial()` function helps us to create 3 important rows in `roles` collection.

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 duplications for `username` and `email`
- check if `roles` in the request is legal or not

middlewares/verifySignUp.js

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

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

```

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

checkDuplicateUsernameOrEmail = (req, res, next) => {
    // Username
    User.findOne({
        username: req.body.username
    }).exec((err, user) => {
        if (err) {
            res.status(500).send({ message: err });
            return;
        }

        if (user) {
            res.status(400).send({ message: "Failed! Username is already in use!" });
            return;
        }

        // Email
        User.findOne({
            email: req.body.email
        }).exec((err, user) => {
            if (err) {
                res.status(500).send({ message: err });
                return;
            }

            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 ${req.body.roles[i]} does not exist!`
                });
                return;
            }
        }
    }
}

```

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

```
next();  
};  
  
const verifySignUp = {  
  checkDuplicateUsernameOrEmail,  
  checkRolesExisted  
};  
  
module.exports = verifySignUp;
```

To process Authentication & Authorization, we create following 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

middlewares/authJwt.js

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

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

```

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

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.findById(req.userId).exec((err, user) => {
        if (err) {
            res.status(500).send({ message: err });
            return;
        }

        Role.find(
            {
                _id: { $in: user.roles }
            },
            (err, roles) => {
                if (err) {
                    res.status(500).send({ message: err });
                    return;
                }

                for (let i = 0; i < roles.length; i++) {
                    if (roles[i].name === "admin") {
                        next();
                        return;
                    }
                }
            }
        );
    });
};

```

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#) Accept

```

    );
  });
};

isModerator = (req, res, next) => {
  User.findById(req.userId).exec((err, user) => {
    if (err) {
      res.status(500).send({ message: err });
      return;
    }

    Role.find(
      {
        _id: { $in: user.roles }
      },
      (err, roles) => {
        if (err) {
          res.status(500).send({ message: err });
          return;
        }

        for (let i = 0; i < roles.length; i++) {
          if (roles[i].name === "moderator") {
            next();
            return;
          }
        }

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

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

```

middlewares/index.js

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

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

```
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

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

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

```

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

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

exports.signup = (req, res) => {
  const user = new User({
    username: req.body.username,
    email: req.body.email,
    password: bcrypt.hashSync(req.body.password, 8)
  });

  user.save((err, user) => {
    if (err) {
      res.status(500).send({ message: err });
      return;
    }

    if (req.body.roles) {
      Role.find(
        {
          name: { $in: req.body.roles }
        },
        (err, roles) => {
          if (err) {
            res.status(500).send({ message: err });
            return;
          }

          user.roles = roles.map(role => role._id);
          user.save(err => {
            if (err) {
              res.status(500).send({ message: err });
              return;
            }

            res.send({ message: "User was registered successfully!" });
          });
        }
      );
    } else {
      Role.findOne({ name: "user" }, (err, role) => {
        if (err) {
          res.status(500).send({ message: err });
        }
      });
    }
  });
}

```

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#) Accept

```

        user.roles = [role._id];
        user.save(err => {
            if (err) {
                res.status(500).send({ message: err });
                return;
            }

            res.send({ message: "User was registered successfully!" });
        });
    });
}

exports.signin = (req, res) => {
    User.findOne({
        username: req.body.username
    })
    .populate("roles", "-__v")
    .exec((err, user) => {
        if (err) {
            res.status(500).send({ message: err });
            return;
        }

        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
        });
    });
}

```

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

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

```

        authorities.push("ROLE_" + user.roles[i].name.toUpperCase());
    }
    res.status(200).send({
        id: user._id,
        username: user.username,
        email: user.email,
        roles: authorities,
        accessToken: token
    });
}
);

```

Controller for testing Authorization

There are 4 functions:

- /api/test/all for public access
- /api/test/user for loggedin users (any role)
- /api/test/mod for **moderator** users
- /api/test/admin for **admin** users

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.");
};

```

Let's combine middlewares with controller functions 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. We use cookies to improve your experience with the site. To find out more, you can read the full (accessing protected resources).

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

Authentication:

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

routes/auth.routes.js

```
const { verifySignUp } = require("../middlewares");
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 forloggedin users (user/moderator/admin)
- GET /api/test/mod for moderator
- GET /api/test/admin for admin

routes/user.routes.js

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

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

```

const { authJwt } = require("../middlewares");
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`.

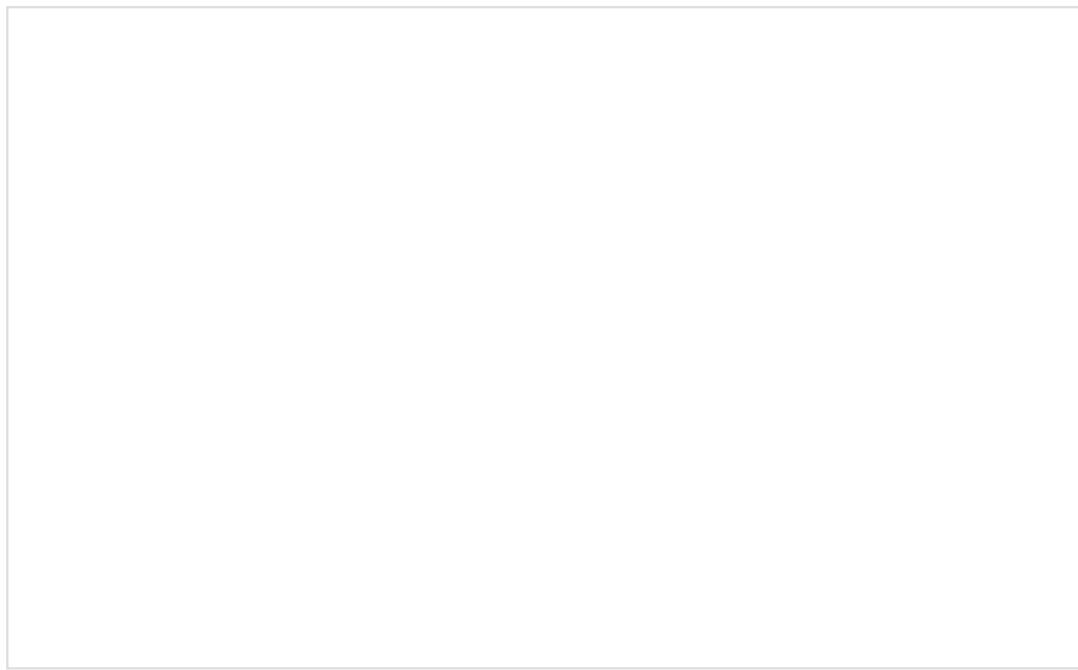
The console shows:

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

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

```
Server is running on port 8080.  
Successfully connect to MongoDB.  
added 'user' to roles collection  
added 'admin' to roles collection  
added 'moderator' to roles collection
```

Let's check roles collection in MongoDB database:

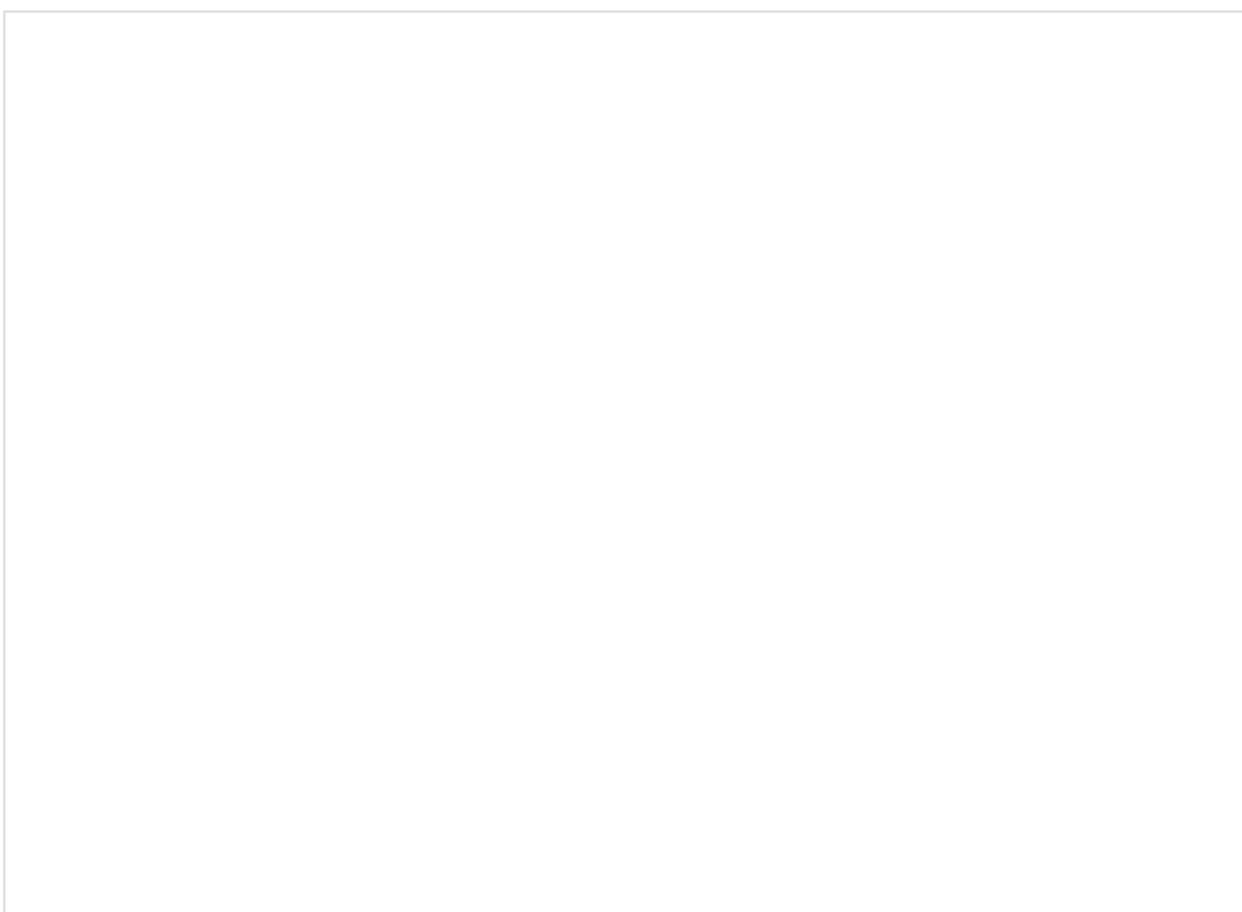


Register some users with /signup API:

- **admin** with admin role
- **modera** with moderator and user roles
- **bezkoder** with user role

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

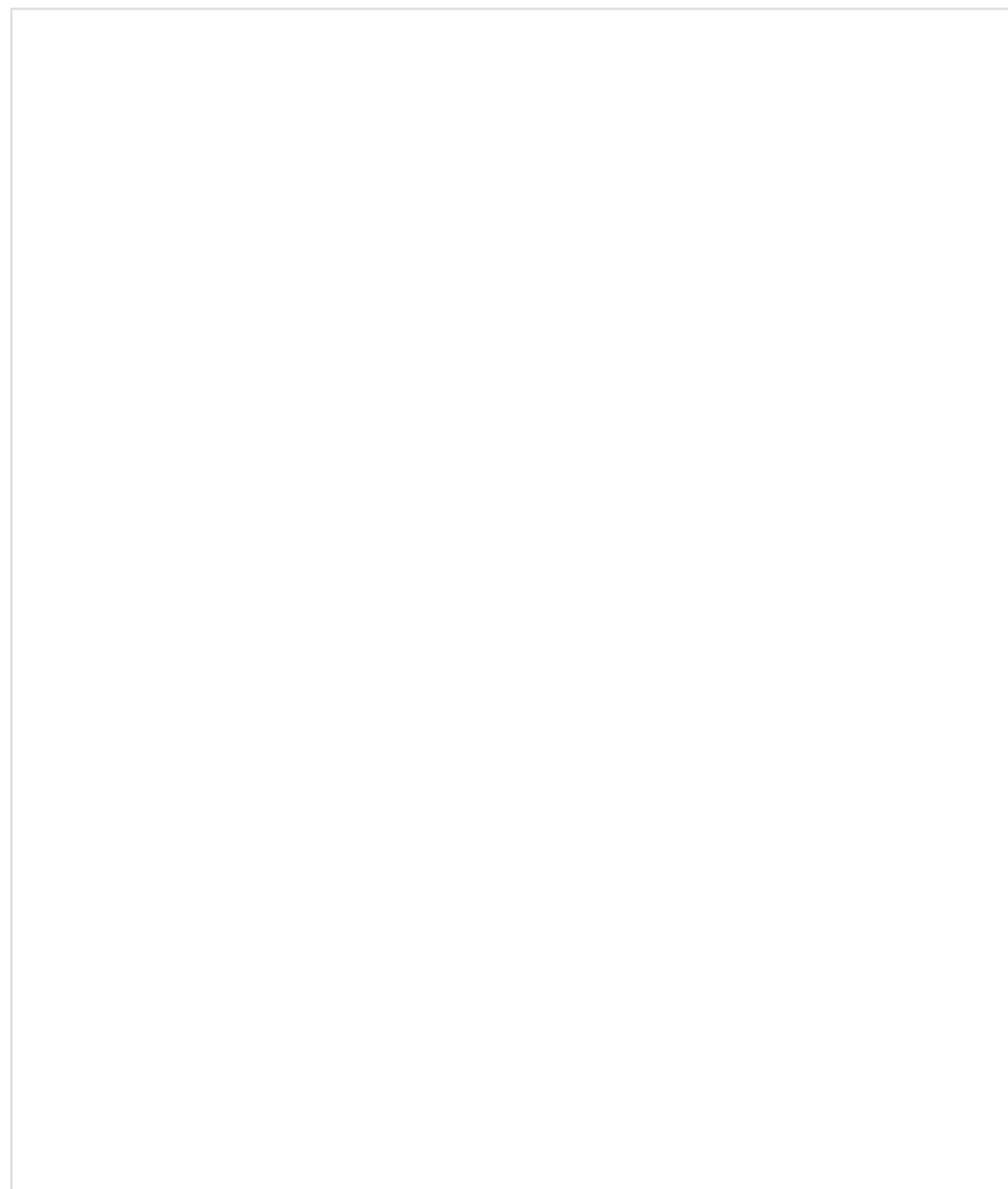
[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/) Accept



users collection after signup could look like this.

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](https://bezkoder.com/privacy-policy/) Accept



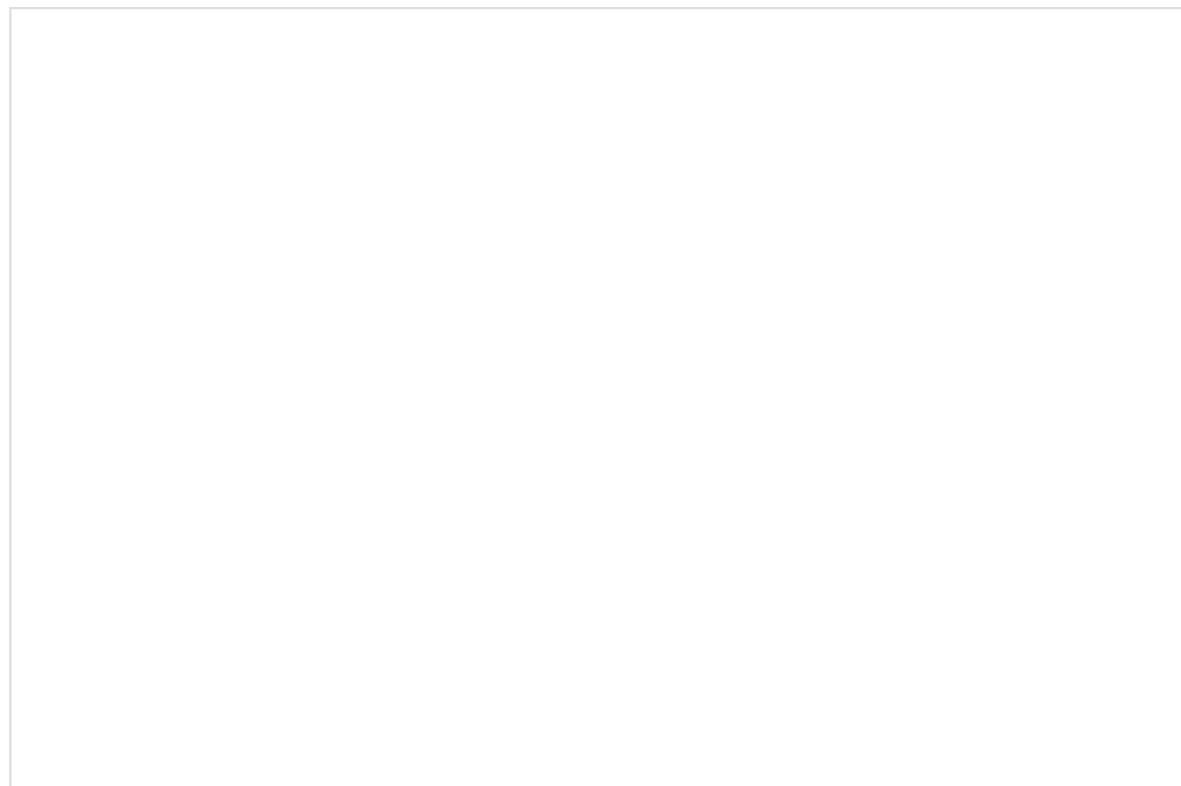
Access public resource: GET /api/test/all

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

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



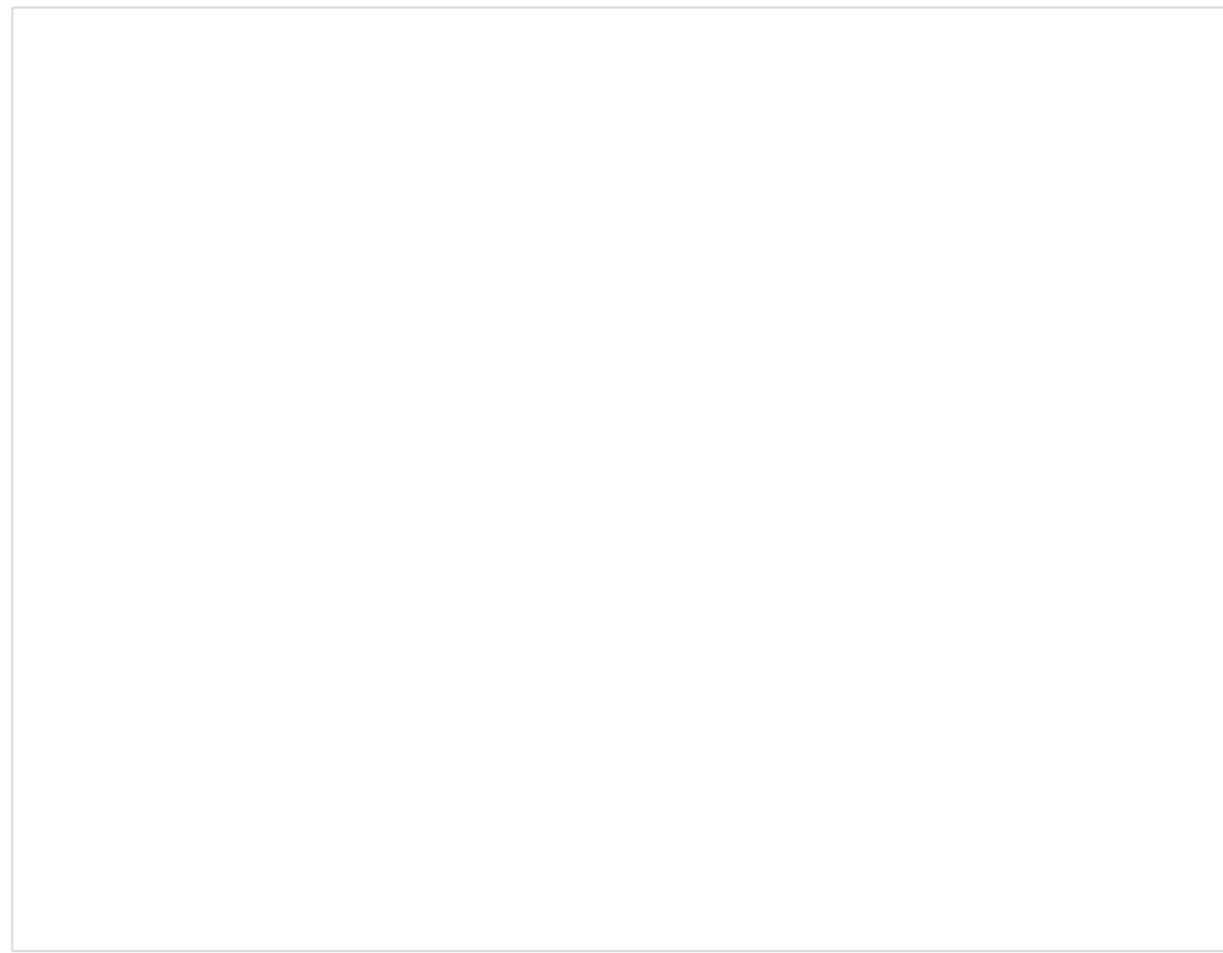
Access protected resource: GET /api/test/user



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

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

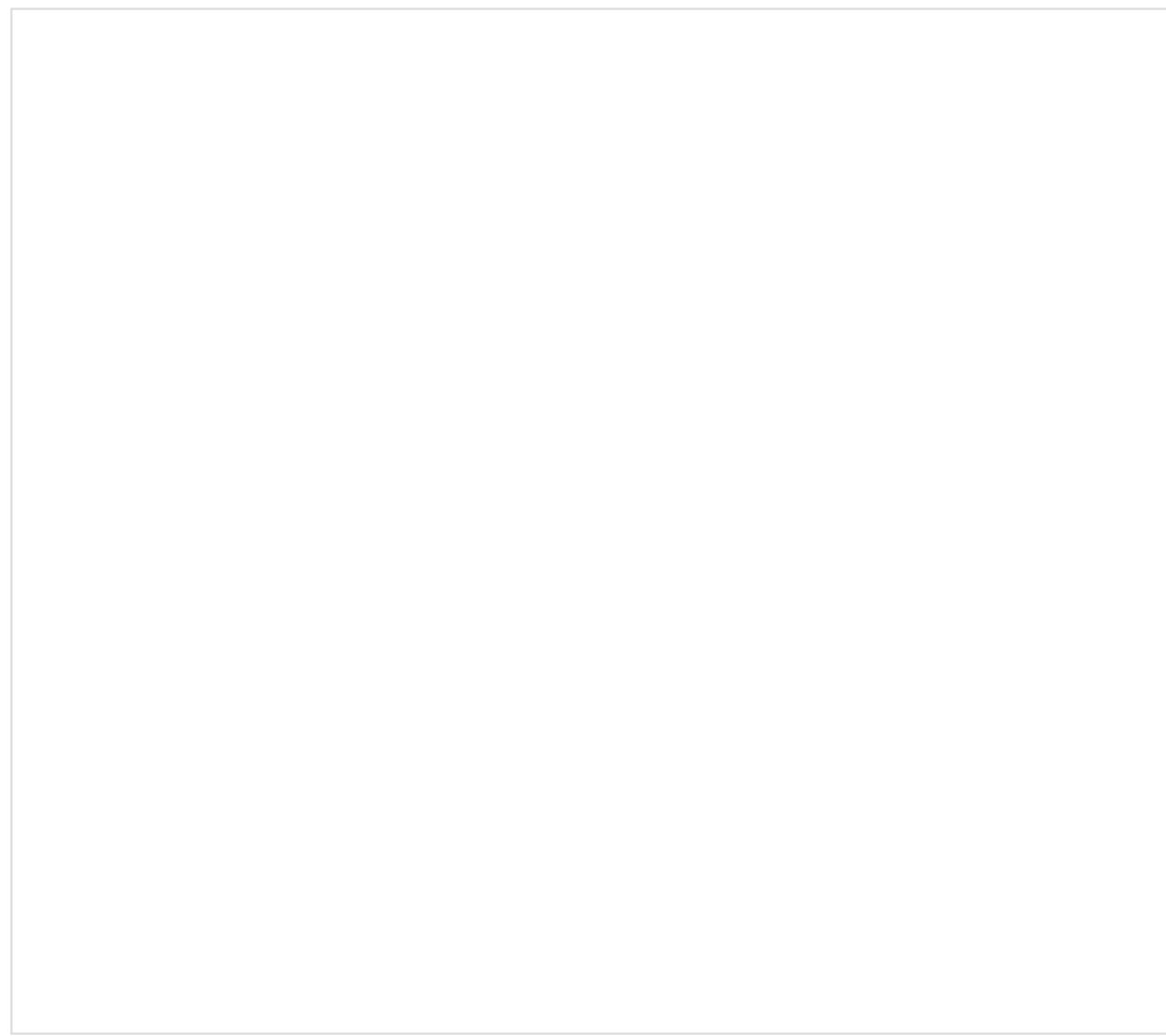
[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/) Accept



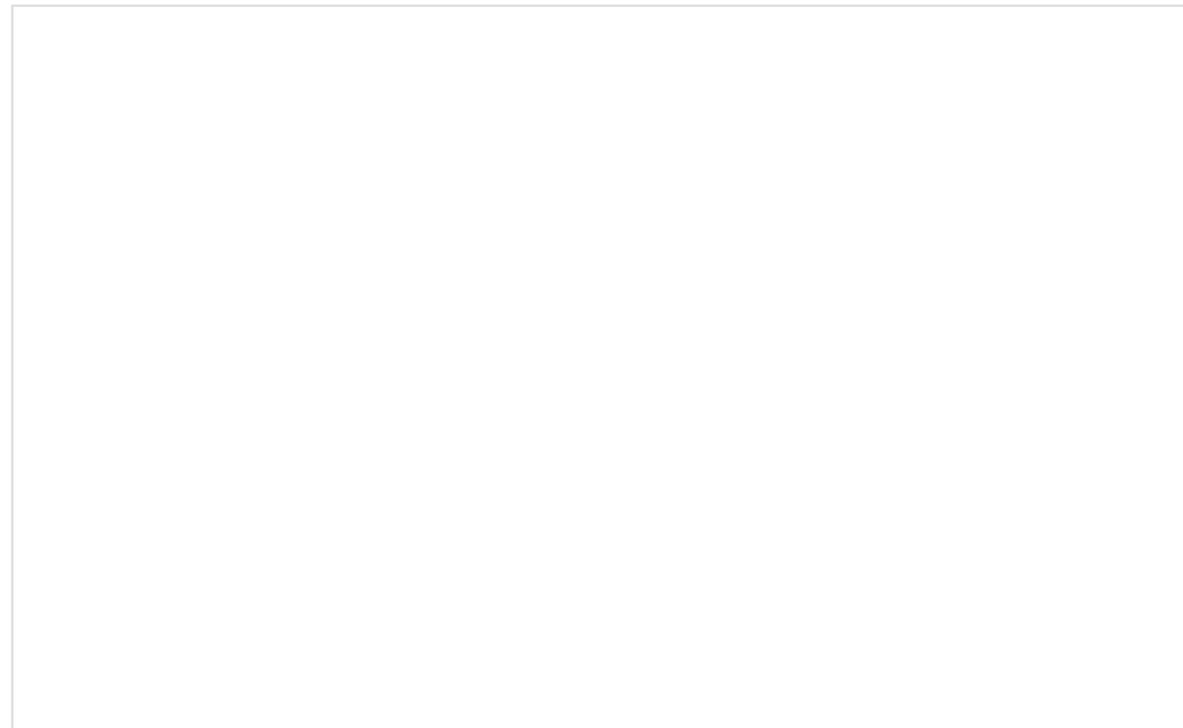
Login an legal account: POST /api/auth/signin

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

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



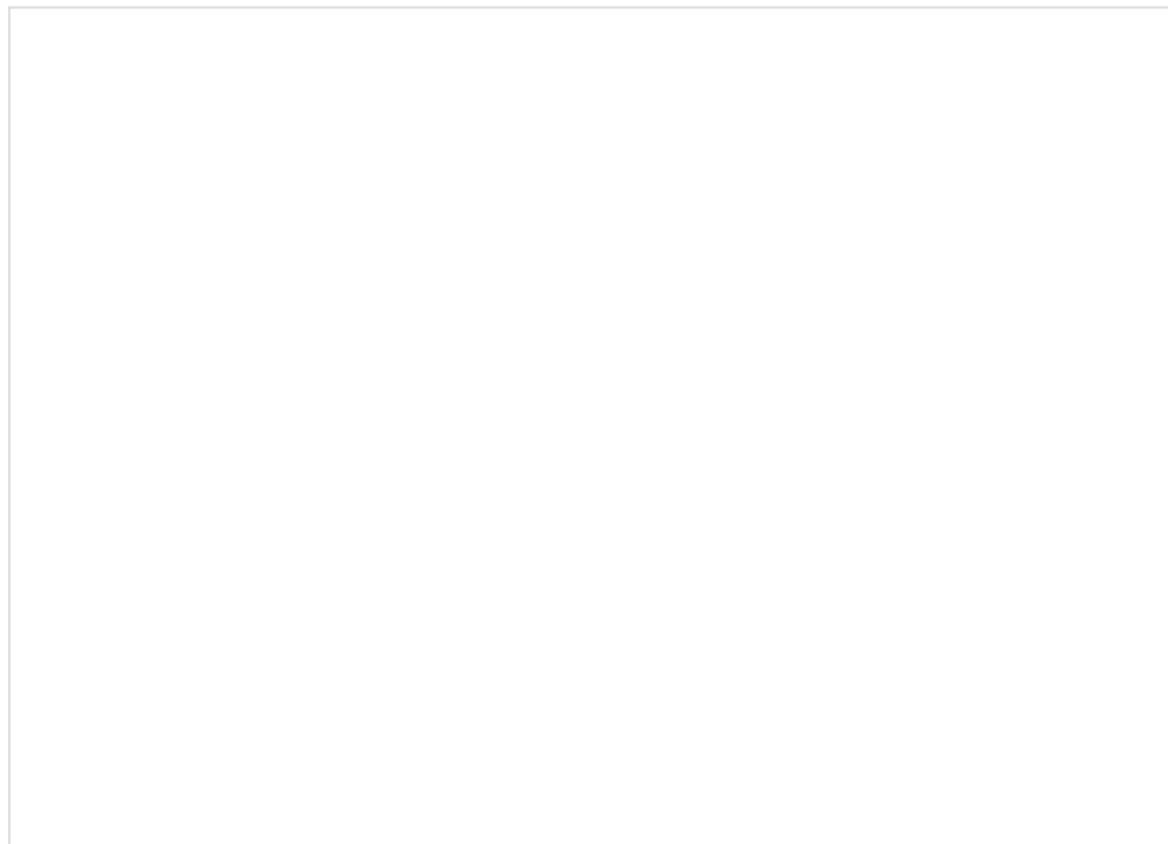
Access protected resources: GET /api/test/user



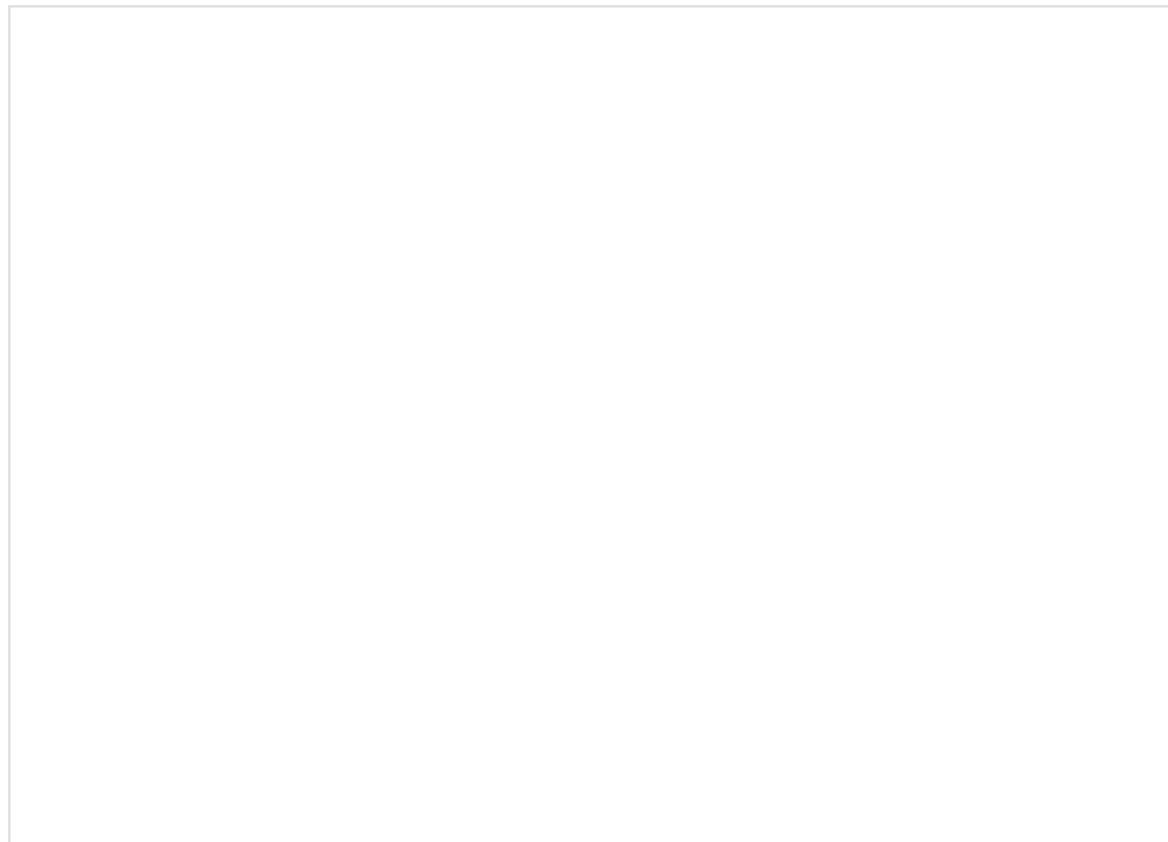
We use cookies to improve your experience with the site. To find out more, you can read the full

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

GET /api/test/admin



If we use a wrong access token:



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

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

Oh yeah! Today we've learned so many interesting things about Node.js MongoDB User Authentication with JWT (JSONWebToken) in just a Node.js Express Rest Api example. You also know how to implement role-based Authorization to restrict access to protected resources.

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/>)
- React JWT Authentication (without Redux) example (<https://bezkoder.com/react-jwt-auth/>)
- React Redux JWT Authentication example (<https://bezkoder.com/react-redux-jwt-auth/>)

Fullstack with React.js: MERN stack Authentication example (<https://bezkoder.com/react-node-mongodb-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>
(<http://expressjs.com/en/guide/routing.html>)
- In-depth Introduction to JWT-JSON Web Token (<https://bezkoder.com/jwt-json-web-token/>)
- <https://mongoosejs.com/docs/queries.html>
(<https://mongoosejs.com/docs/queries.html>)
- <https://mongoosejs.com/docs/api/model.html>
(<https://mongoosejs.com/docs/api/model.html>)

Source Code

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

authentication (<https://bezkoder.com/tag/authentication/>)

authorization (<https://bezkoder.com/tag/authorization/>) express (<https://bezkoder.com/tag/express/>)

jwt (<https://bezkoder.com/tag/jwt/>) mongodb (<https://bezkoder.com/tag/mongodb/>)
We use cookies to improve your experience with the site. To find out more, you can read the full
mongoose (<https://bezkoder.com/tag/mongoose/>) node.js (<https://bezkoder.com/tag/node-js/>)
[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](https://bezkoder.com/privacy-policy/) Accept
security (<https://bezkoder.com/tag/security/>)

71 thoughts to “Node.js + MongoDB: User Authentication & Authorization with JWT”

dasuksta

March 24, 2020 at 1:17 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-619>)

Hello, thanks for your work on the guide...

Having an issue connecting to mongoDB, fixed the connection by adding:

const dbConfig = require("./config/db.config"); to server JS – the \${dbConfig.HOST} variables can be found...

but cant figure were the function initial()... code is placed? the connection just hangs.

Warm Regards

[REPLY](#)

bezkoder

March 25, 2020 at 1:32 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-629>)

Hi, it is in *server.js*. You can find where to put it in the tutorial, or source code at Github (<https://github.com/bezkoder/node-js-jwt-auth-mongodb/blob/master/server.js>).

[REPLY](#)

dasuksta

March 27, 2020 at 10:26 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-665>)

My bad! was chasing a spelling error, doh!

Thanks for the reply 😊

[REPLY](#)

vince

June 20, 2020 at 1:32 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3375>)

We use cookies. Rightfully following experience with the configToFind out more about config, this full missing in Initialize Mongoose section.

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](https://bezkoder.com/privacy-policy/)

Accept

[REPLY](#)

dasuksta

March 27, 2020 at 10:24 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-663>)

Hello, thank for the great guide – I have followed it closely, had a few issues...

Minor:

1. In the your project structure, you defined the folder “middleware” – but in the guide is referred to as “middlewares”

2. Initialize Mongoose – adding to the server.js file the db configurations were not found – I added:

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

just above the mongoose config – worked!

Having a major issue, when starting up the app and using Postman to register a new user, get the following error serverside:

`TypeError: Cannot read property 'findOne' of undefined`

Double check the code and tried looking on-line, couldn't find anything solid to go on – thought the feedback may help. Thank you

[REPLY](#)

Rafael

May 21, 2020 at 3:26 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-2710>)

Hey check your

`auth.controller.js` and `index.js` to make sure you have correct values there...

I had `db.users = require("./user.model.js")(mongoose);` in `index.js` and `const User = db.user;` in the controller.

Please notice the typo there...

[REPLY](#)

Varun Indoria

April 19, 2020 at 7:02 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-946>)

Nice tutorial!! Really helpful for me.. Thanks a lot!!

We use cookies to improve your experience with the site. To find out more, you can read the full [REPLY](#)

[Privacy & Policy](#) (<https://bezkoder.com/privacy-policy/>)

Accept

Jimoh Rildwan

April 27, 2020 at 12:13 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-1670>)

Great tutorial. Worked perfectly for me...Really helpful man

REPLY

Apurva

May 5, 2020 at 7:05 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-1972>)

Hi I am getting error as app.use is not a function. Can anyone please help?

REPLY

bezkoder

May 5, 2020 at 2:10 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-1999>)

Hi, please make sure that you initialize Express correctly.

REPLY

Andrea

May 6, 2020 at 6:59 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-2037>)

Hey thank you, everything worked perfectly the first time! 😊

FYI – I'm on Windows, and I have previously been instructed to use bcryptjs, but fyi, plain old bcrypt works just fine. I already had bcrypt installed, so instead of also installing bcryptjs, I decided to just try and see if it works, and it did!

I truly appreciate when I come across a well-written article from someone who actually has working code lol... It's frustrating how many people write articles and clearly did not even use their own code... grr... so, thanks again!

REPLY

basha

May 9, 2020 at 4:19 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-2152>)

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](https://bezkoder.com/privacy-policy/) Accept

I got this error when model creation, exactly got this one in this line db.user = require("./user.model");
 message: 'Cannot overwrite `users` model once compiled.',
 name: 'OverwriteModelError'. how to reproduce this one.

[REPLY](#)**bezkoder**

May 10, 2020 at 8:38 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-2188>)

Hi, maybe you have instantiated `mongoose.Model()` on the same schema twice.

[REPLY](#)**Rafael**

May 12, 2020 at 2:10 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-2331>)

Great tutorial!

This is something I was looking for.

You could mention in your other tutorials, this could be used as small reference for a one-to-many for NodeJS – Express – MongoDB

[REPLY](#)**rami**

May 14, 2020 at 4:21 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-2433>)

Great tutorial

i got this error "TypeError: Cannot read property 'username' of undefined
 at checkDuplicateUsernameOrEmail" can anyone please help?

[REPLY](#)**bezkoder**

May 15, 2020 at 8:11 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-2450>)

Hi, maybe you forgot to set `'Content-Type': 'application/json'` in the HTTP We use cookies to improve your experience with the site. To find out more, you can read the full request header.

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#)

Accept

[REPLY](#)

harshita b patil

September 2, 2020 at 11:15 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4578>)

thank you very much for your guide
it will be great help if help me out in solving this prblm

error is:

```
C:\Users\Lenovo\feb-mern-2020\mern project\node-js-jwt-auth-
mongodb\server.js:24
  .connect(`mongodb://${dbConfig.HOST}:${dbConfig.PORT}/${dbConfig.DB}`, {
    ^

```

```
TypeError: Cannot read property 'connect' of undefined
at Object. (C:\Users\Lenovo\feb-mern-2020\mern project\node-js-jwt-auth-
mongodb\server.js:24:4)
at Module._compile (internal/modules/cjs/loader.js:1158:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:1178:10)
at Module.load (internal/modules/cjs/loader.js:1002:32)
at Function.Module._load (internal/modules/cjs/loader.js:901:14)
at Function.executeUserEntryPoint [as runMain]
(internal/modules/run_main.js:74:12)
at internal/main/run_main_module.js:18:47
```

[REPLY](#)

mako

February 21, 2021 at 2:35 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-7149>)

you need [const dbConfig = require("./app/config/db.config.js");] in server.js

[REPLY](#)

Nima

May 21, 2020 at 9:22 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-2736>)

you must move these lines

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

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#) [Accept](#)

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

in server.js file.
```

[REPLY](#)**Blade**

July 23, 2020 at 9:47 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3908>)

I was having the same issue. I'm not sure why, but node was having an issue with the import of that verifySignup module. I had to change my import to explicitly be:

```
const verifySignup = require('../middleware/verifySignUp.js')

rather than trying to use the neater importing/object destructuring used in the tutorial.
```

[REPLY](#)**Jon**

April 3, 2021 at 7:56 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-7891>)

Check the imports and exports. Do not use const { verifySignUp } = require("../middlewares"); instead use const verifySignUp = require("../middlewares");

[REPLY](#)**jm61**

June 18, 2020 at 2:06 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3321>)

Hello there,

I got a problem with the initial() function, result is "Role.estimatedDocumentCount is not a function"

I'm using mongoDB on cloud Atlas and mongoose version is 5.9.19,

Thank you for debug information, I really want to practice more this wonderful tutorial 😊

Cheers.

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#)

[Accept](#)

jm61

June 18, 2020 at 8:22 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3333>)

it's working now! Great Tutorial thanks

[REPLY](#)

yashu

June 27, 2020 at 11:10 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3481>)

Role.estimatedDocumentCount is not a function" I got this issue due to old version of mongoose
used Role.collection.estimatedDocumentCount to solve this.

[REPLY](#)

Partha Chakraborty

June 23, 2020 at 11:07 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3413>)

got this error "TypeError: Cannot read property 'username' of undefined
at checkDuplicateUsernameOrEmail" can anyone please help?

Same is here also in postman I have set 'Content-Type': 'application/json' in the HTTP request header. inspite of that I am getting this error please helpme.

[REPLY](#)

Tonho da Lua

November 20, 2020 at 3:15 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5478>)

you probably have your routes set before body-parser parameters

[REPLY](#)

jm61

June 25, 2020 at 12:02 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3448>)

Hello Bezkoder,

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

Role (user, moderator or admin)?

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#)

Accept

[REPLY](#)**harshita b patil**September 2, 2020 at 11:18 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4580>)

Hey jm61 even i want to know this, did you get how to redirect ?

[REPLY](#)**HM**September 2, 2020 at 1:16 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4582>)

That would be of interest for me, too 😊

[REPLY](#)**Jon**April 3, 2021 at 7:57 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-7892>)

[authJwt.verifyToken, authJwt.isModerator, authJwt.isAdmin]

[REPLY](#)**Juan G Valdes**June 30, 2020 at 2:58 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3545>)

Hi, loved the post I was wondering if you had any insides on how you would go about unit testing your middleware classes I was looking at verifySignUp, and haven't been able to figure out how to mock the DB in order to isolate any of the methods for unit testing.

Thanks a lot for your post!

[REPLY](#)**james**We use cookies to improve your experience with the site. To find out more, you can read the full [Privacy & Policy](#) (<https://bezkoder.com/privacy-policy/>)
July 10, 2020 at 1:53 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3680>)[Privacy & Policy](#) (<https://bezkoder.com/privacy-policy/>)

Accept

```
module.exports = {
  HOST: "localhost",
  PORT: 27017,
  DB: "bezkoder_db"
};
```

to my mongo db cloud url , i am getting an error after changing it to this

```
module.exports = {
  url: "mongodb+srv://james:success@main.egoxz.azure.mongodb.net/crud?
  retryWrites=true&w=majority"
};
```

the error i got is this

```
throw new MongooseError('The `uri` parameter to `openUri()` must be a ' +
^
```

MongooseError: The `uri` parameter to `openUri()` must be a string, got
 "undefined". Make sure the first parameter to `mongoose.connect()` or
 `mongoose.createConnection()` is a string.

```
at NativeConnection.Connection.openUri (C:\Users\user\Documents\node-js-jwt-
auth-mongodb-master\node_modules\mongoose\lib\connection.js:582:11)
```

```
at Mongoose.connect (C:\Users\user\Documents\node-js-jwt-auth-mongodb-
master\node_modules\mongoose\lib\index.js:335:15)
```

```
at Object. (C:\Users\user\Documents\node-js-jwt-auth-mongodb-
master\server.js:24:2)
```

```
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
```

[REPLY](#)

Gustavo

October 18, 2020 at 10:16 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5100>)

James I did like that... in the server.js

I clear:

```
"db.mongoose
```

```
.connect(`mongodb://${dbConfig.HOST}:${dbConfig.PORT}/${dbConfig.DB}` , {
```

```
useNewUrlParser: true,
```

```
useUnifiedTopology: true
```

We use cookies to improve your experience with the site. To find out more, you can read the full [Privacy & Policy](#).

[Privacy & Policy](#) (<https://bezkoder.com/privacy-policy/>) [Accept](#)

```
"const path = 'mongodb+srv://USER:PASSWORD@cluster0.icnc6.mongodb.net/DB?  
retryWrites=true&w=majority';  
db.mongoose  
.connect(path, { useNewUrlParser: true, useCreateIndex: true, useUnifiedTopology:  
true  
})"
```

[REPLY](#)**Kristoffer J**

July 21, 2020 at 10:02 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3873>)

Great tutorial and great tutorial site!

I found a weird bug though, might be caused by newer versions installed here.

auth.controller.signup: Role.Find({ name: {\$in: req.body.roles} } is case sensitive and matches what is stored in db.

verifySignup.checkRoleexisted: if(!ROLES.includes(req.body.roles[i])) is also case sensitive and require lowercase because the ROLES are returned as lowercase.

I solved it by changing to "if(!ROLES.includes(req.body.roles[i].toLowerCase()))" but it is not beautiful in any way.

[REPLY](#)**Olaitan**

July 26, 2020 at 7:17 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-3965>)

I am getting this error in my postman “Client network socket disconnected before secure TLS connection was established” what can I do, please help

[REPLY](#)**fati**

July 28, 2020 at 12:34 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4007>)

thank you very much

can you help me, How to extract id user I need just id

[REPLY](#)

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#)

Accept

Dan

August 2, 2020 at 2:01 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4144>)

Hi, great article I must say. Think it's clicked now how JWT and Express.js working so thank you very much!

All is working great, already building onto the system.

Just wondering, what would be the best way to logout the user, reset the password, forgot password? Any tips, please?

Keep up the good work

[REPLY](#)

Chitransh Popli.

August 22, 2020 at 12:02 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4424>)

Yeah have the same doubt.

(And same feeling about the article too. Very helpful).

[REPLY](#)

Jennifer

August 9, 2020 at 2:16 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4243>)

I have a question – you have an array of functions being passed to an express route. I wasn't aware you could do this – and my version of express only allows a single function (not an array). Do you have documentation on how this will execute?

[REPLY](#)

bezkoder

August 9, 2020 at 3:33 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4244>)

Hi, you can find the instruction here:

<https://expressjs.com/en/guide/routing.html>

(<https://expressjs.com/en/guide/routing.html>) (Route handlers)

[REPLY](#)

We use cookies to improve your experience with the site. To find out more, you can read the full [Privacy & Policy](#). March 8, 2021 at 9:09 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-7482>)

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#) [Accept](#)

If you have it like me, then the following worked for me because there can be any amount of callback functions sent and I found out that the first callback is executed first:

```
router.get('/test/admin', authJwt.verifyToken, authJwt.isAdmin, adminBoard)
```

[REPLY](#)

Salman Kazi

August 14, 2020 at 8:08 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4307>)

How To Redirect From Backend To Frontend After Authentication ?

[REPLY](#)

Asli

August 20, 2020 at 5:41 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4394>)

This is awesome. it will be much helpful if you can guide for APIs s in clouddb.

[REPLY](#)

Ironlord

August 27, 2020 at 6:37 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4484>)

Great tutorial. I really enjoyed reading it.

Would be nice to have fully asynchronous functions.

[REPLY](#)

Noemi

September 1, 2020 at 6:33 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4574>)

Hello,

I am having a lot of issues with this tutorial and nothing is working for me so far...

I follow the Node.js, Express & MongoDB: Build a CRUD Rest Api example to get a front and back end. I'm basically trying to do this tutorial to get the user to log in to then access all the Tutorials (from the mentioned tutorial). But it wasn't working. Now

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

User.findOne is not a function at checkDuplicateUsername

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](https://bezkoder.com/privacy-policy/)

Accept

I am on my mac, using mongo community 4.4. I'm not sure why I am getting this error

[REPLY](#)**Akash**

September 6, 2020 at 1:38 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4612>)

```
C:\Users\test\Desktop\testing\node-js-jwt-auth-mongodb\server.js:24
  .connect(`mongodb://${dbConfig.HOST}:${dbConfig.PORT}/${dbConfig.DB}` , {
    ^

```

```
TypeError: Cannot read property 'connect' of undefined
at Object. (C:\Users\test\Desktop\testing\node-js-jwt-auth-mongodb\server.js:24:4)
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
```

getting an error like this please help.

[REPLY](#)**phuongle**

September 10, 2020 at 3:11 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4669>)

you can check again module.exports in config file! I think the problem about the syntax :))

[REPLY](#)**mako**

February 21, 2021 at 2:34 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-7147>)

you need [const dbConfig = require("./app/config/db.config.js");] in server.js

[REPLY](#)

Nick
We use cookies to improve your experience with the site. To find out more, you can read the full
September 9, 2020 at 12:27 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4650>)

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#) [Accept](#)

Upon downloading the code from your GitHub page the code does not work. I get the error Cannot GET /api/auth/signup. Can you advise how to fix this as I did not modify the code in any way.

[REPLY](#)**bezkoder**

September 9, 2020 at 1:28 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4651>)

Hi, please make sure that your MongoDB runs correctly with db configuration, and the Node.js server runs successfully also.

[REPLY](#)**Zahidul Islam**

September 22, 2020 at 10:44 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-4815>)

Really awesome.

[REPLY](#)**Ivan Goh**

October 14, 2020 at 12:42 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5033>)

Great Guide! Just a slight problem on my end with the Bcrypt hashing as it continuously gives me an error that writes : "Error: Illegal arguments: undefined, string at Object.bcrypt.hashSync". Not sure if this is normal or dependency error, please advice. Thanks in advance!

[REPLY](#)**mui**

January 23, 2021 at 10:59 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-6466>)

you need to set the dropdown to JSON in you postman for the body parameter

[REPLY](#)

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

[Privacy & Policy](#) (<https://bezkoder.com/privacy-policy/>) [Accept](#)

Donpius

February 23, 2021 at 2:56 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-7187>)

I've the same issue, too.

Have anybody a solution?

REPLY

Donpius

February 23, 2021 at 3:01 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-7188>)

okay, i fixed it.

Solution was: a typo

REPLY

Malik Iyanda

April 26, 2021 at 2:57 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-8286>)

a typo where cause i cant find any in mine

REPLY

Sanfer Noronha

October 14, 2020 at 4:40 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5037>)

It is giving error for an HTTPS request....working for http. Why?

REPLY

juan carlos

October 18, 2020 at 3:02 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5106>)

Muy buen tutorial!!! Gracias

REPLY

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#)

Accept

October 25, 2020 at 6:14 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5191>)

did the x-access-token came as a header or is it typed by copying accessToken and pasting it in headers. plllsssss reply

[REPLY](#)**Adam**

October 29, 2020 at 8:01 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5234>)

Hi sir, that a great tutorial! But i have an error at:

TypeError: Role.estimatedDocumentCount is not a function

Why? Thanks

[REPLY](#)**Crav**

November 23, 2020 at 4:31 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5519>)

I appreciate, cause I found the tutorial what I was looking for. You've ended my four day long hunt! God Bless you man. Have a great day. Thanks!

[REPLY](#)**Ivarag**

December 14, 2020 at 11:16 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5783>)

Hey there,

I really appreciate your site and the code you're offering us.

I'm a (backend)beginner and really blown away from the complexity of your code (in a positive way). It seems so clean and well made. My question is... what is your experience and how long did you code? As far as I can understand this code.. this is clean and clear at the same time! My biggest concern right now is,that I cannot write this whole code alone at all. Backend is special for itself. What is your opinion about that? Thanks in advance!

[REPLY](#)

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

[yagi](#)

[Privacy & Policy](#) (<https://bezkoder.com/privacy-policy/>)

[Accept](#) (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5887>)

Good tutorial! Thanks!

REPLY

mikiyasdaniel

December 28, 2020 at 11:45 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5951>)

make me happy after competing this module...that make me feel i am expert with node.js
....thanks a lot..

REPLY

Alin (<http://alin-andrei.com>)

December 29, 2020 at 3:20 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5973>)

Hi,

First of all thanks for the tutorial. At a level of concepts, I got it. At the practical level, User.findOne is not a function hits me from checkDuplicateUsernameOrEmail() in verifySignUp middleware.

What is this error actually telling me? I am understanding this as the seat is taken somehow.

I have googled it for a while but there's nothing wrong with the syntax from what I can understand. Any solutions or at least a more detailed explanation of why this is occurring?

I have checked that mongoose it's installed. I manage to connect to the mongo database and inser the roles data, but when I try to add a new user via the api i receive this error

TypeError: User.findOne is not a function
at checkDuplicateUsernameOrEmail (/home/alin/Work/BackEnd/token-based-auth/app/middlewares/verifySignUp.js:7:8)

Thanks.

Best,

Alin

REPLY

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#) Accept

All

December 29, 2020 at 7:03 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-5982>)

In my user.model.js at the very last line i was exporting module.exports = 'User'; instead of module.exports = User;

Thanks again!

[REPLY](#)

Shoaib

January 30, 2021 at 11:29 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-6680>)

Thanks for this amazing post, Initially I faced issue but with going through comments. Everything gone fine. I really appreciate your work. Including this blog I found your another blogs useful too. Thanks mate

[REPLY](#)

vir

March 7, 2021 at 4:26 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-7459>)

hello, when I do this in my routes :

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

I have the error

```
/var/www/html/MERN-authentication-with-nodeJS-express-mongo-DB-react-
redux/backend/routes/user.routes.js:14
const { authJwt } = require("../middlewares");
^^^^^
```

SyntaxError: Unexpected token 'const'

[REPLY](#)

Malik Iyanda

March 30, 2021 at 4:19 am (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-7806>)

hi i keep getting network error when trying to submit a form not sure why no error and all is working fine.

We use cookies to improve your experience with the site. To find out more, you can read the full [REPLY](#)

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](#) [Accept](#)

nanous

April 26, 2021 at 9:14 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-8287>)

hiii thank you so much for this tutorial but i'm actually facing some problems with the roles as they don't show in the db when i sign up a new user

[REPLY](#)**Coder**

May 3, 2021 at 10:23 pm (<https://bezkoder.com/node-js-mongodb-auth-jwt/#comment-8370>)

Hello, I want to display the data of a user in flatlist , but i didn't find what to modify in routes/controller so i can do this , i'll appreciate if you can help me

[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.

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

[Privacy & Policy](#) (<https://bezkoder.com/privacy-policy/>) [Accept](#)

[◀ React JWT Authentication \(without Redux\) example \(https://bezkoder.com/react-jwt-auth/\)](https://bezkoder.com/react-jwt-auth/)[React Pagination using Hooks example ➤ \(https://bezkoder.com/react-pagination-hooks/\)](https://bezkoder.com/react-pagination-hooks/)

CATEGORIES

- 📁 Android (<https://bezkoder.com/category/android/>)
- 📁 Angular (<https://bezkoder.com/category/angular/>)
- 📁 Courses (<https://bezkoder.com/category/courses/>)
- 📁 Dart (<https://bezkoder.com/category/dart/>)
- 📁 Deployment (<https://bezkoder.com/category/deployment/>)
- 📁 Django (<https://bezkoder.com/category/django/>)
- 📁 Firebase (<https://bezkoder.com/category/firebase/>)
- 📁 Flutter (<https://bezkoder.com/category/flutter/>)
- 📁 Full Stack (<https://bezkoder.com/category/full-stack/>)
- 📁 GraphQL (<https://bezkoder.com/category/graphql/>)
- 📁 Java (<https://bezkoder.com/category/java/>)
- 📁 Javascript (<https://bezkoder.com/category/javascript/>)
- 📁 Kotlin (<https://bezkoder.com/category/kotlin/>)
- 📁 MongoDB (<https://bezkoder.com/category/mongodb/>)
- 📁 Node.js (<https://bezkoder.com/category/node-js/>)
- 📁 Python (<https://bezkoder.com/category/python/>)
- 📁 React (<https://bezkoder.com/category/react/>)
- 📁 Reactor (<https://bezkoder.com/category/reactor/>)
- 📁 Security (<https://bezkoder.com/category/security/>)
- 📁 Spring (<https://bezkoder.com/category/spring/>)
- 📁 Testing (<https://bezkoder.com/category/testing/>)
- 📁 Vue.js (<https://bezkoder.com/category/vue/>)

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

[Privacy & Policy \(<https://bezkoder.com/privacy-policy/>\)](https://bezkoder.com/privacy-policy/)[Accept](#)

FOLLOW US



(htt

ps://

WW

w.yo

utub

e.co

m/c

han

f nel/

(htt UCp (htt

ps:// 0mx ps://

face 9RH gith

boo 0Jxa ub.c

k.co Fsm om/

m/b MvK bezk

ezko XA8 oder

der) 6Q))

TOOLS

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

We use cookies to improve your experience with the site. To find out more, you can read the full [DMCA PROTECTED](https://www.dmca.com/Protection/Status.aspx?ID=3f543dd5-c6d8-4208-9a6b-0e92057fd597&return=https://bezkoder.com/node-js-mongodb-auth-jwt/) (<https://www.dmca.com/Protection/Status.aspx?ID=3f543dd5-c6d8-4208-9a6b-0e92057fd597&return=https://bezkoder.com/node-js-mongodb-auth-jwt/>)

Accept

[Home \(https://bezkoder.com/\)](https://bezkoder.com/) [Privacy Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/)

[Contact Us \(https://bezkoder.com/contact-us/\)](https://bezkoder.com/contact-us/) [About Us \(https://bezkoder.com/about/\)](https://bezkoder.com/about/)

BezKoder 2019

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

[Privacy & Policy \(https://bezkoder.com/privacy-policy/\)](https://bezkoder.com/privacy-policy/) [Accept](#)