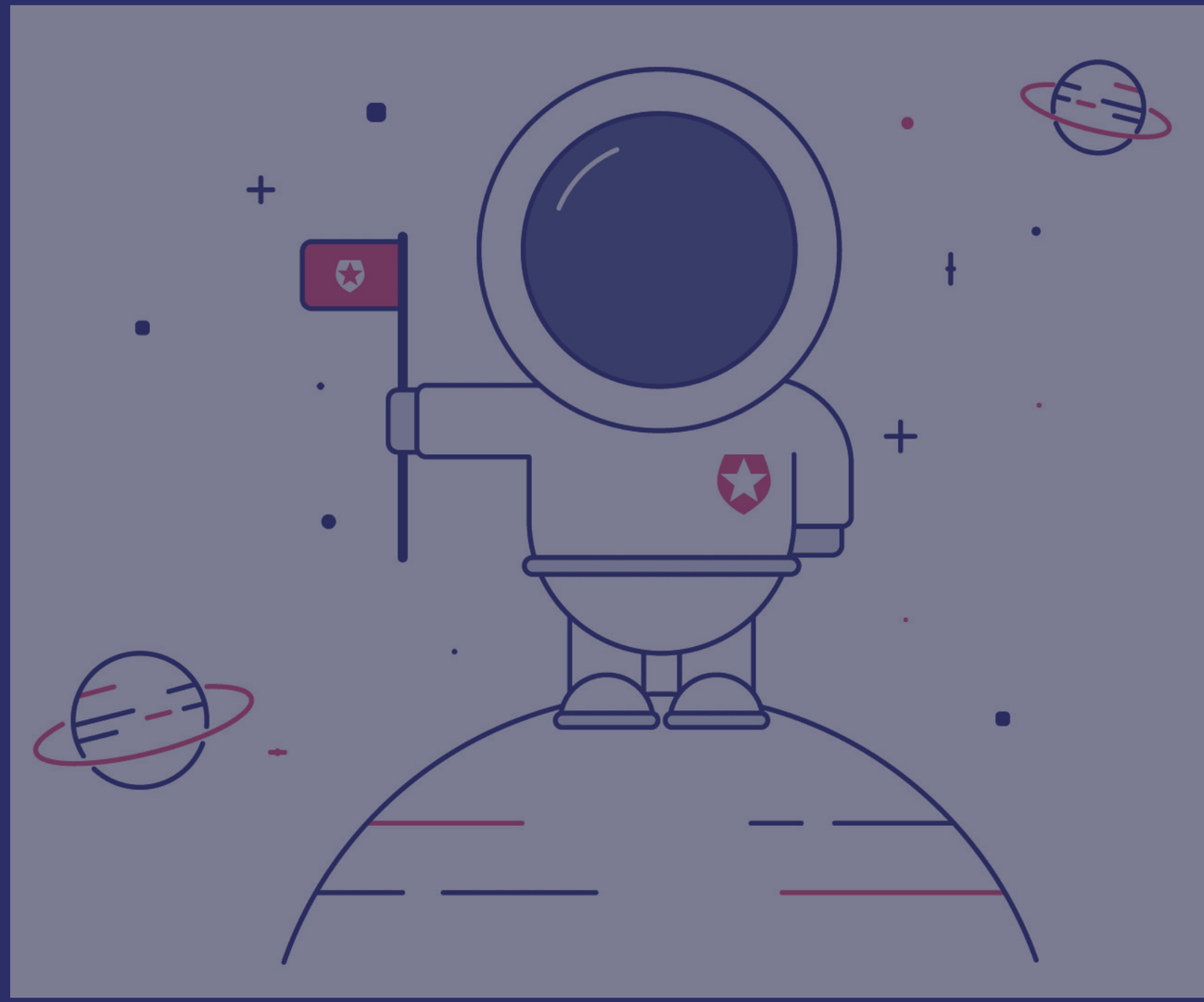
Securing NodeJS APIs with JWT

Who am I

Chathu Vishwajith

Auth0 Ambassador



Why this example ~~has~~ needs database and encryption?

Securing NodeJS APIs with JWT

Your Node.js authentication tutorial is (probably) wrong

tl;dr: I went on a search of Node.js/Express.js authentication tutorials. All of them were incomplete or made a security mistake in some way that can potentially hurt new users. This post explores some common authentication pitfalls, how to avoid them, and what to do to help yourself when your tutorials don't help you anymore. I am still searching for a robust, all-in-one solution for authentication in Node/Express that rivals Rails's Devise.

> ***Update (Aug 7)****: RisingStack has reached out and no longer stores passwords in plaintext in their tutorial, opting to move to bcrypt in their example codes and tutorials.*
> ***Update (Aug 8):*** *Editing title to* Your Node.js authentication tutorial is (probably) wrong*, as this post has improved some of these tutorials.*
> ***Update (Aug 10):*** Dan McGhan found that one of the tutorials has addressed an issue that I had somehow missed in this documentation. I've omitted the graf for now, as Medium doesn't allow for strikethrough. After all, I make mistakes, too. 😊 An addendum is placed at the end of this article.

In my spare time, I've been digging through various Node.js tutorials, as it seems that every Node.js developer with a blog has released their own tutorial on how to do things *the right way*, or, more accurately, *the way they do them*. Thousands of front-end developers being thrown into the server-side JS maelstrom are trying to piece together actionable knowledge from these tutorials, either by cargo-cult-copypasta or gratuitous use of *npm install* as they scramble frantically to meet the deadlines set for them by outsourcing managers or ad agency creative directors.

What are the dependacies of this project?

Express

Fast, unopinionated, minimalist web framework for Node.js

Mongoose

Elegant mongodb object modeling for node.js

Helmet

Helmet helps you secure your Express apps by setting various HTTP headers

CORS

Express middleware that can be used to enable CORS with various options.

body-parser

Parse incoming request bodies in a middleware before your handlers, available under the `req.body` property.

content-filter

provides protection against NoSQL injection attacks for NodeJS applications

jsonwebtoken

An implementation of JSON Web Tokens for NodeJS

express-jwt

Middleware that validates JsonWebTokens and sets `req.user`

What are the dev dependancies

gulp

gulp is a toolkit for automating painful or time-consuming tasks in your development workflow, so you can stop messing around and build something.

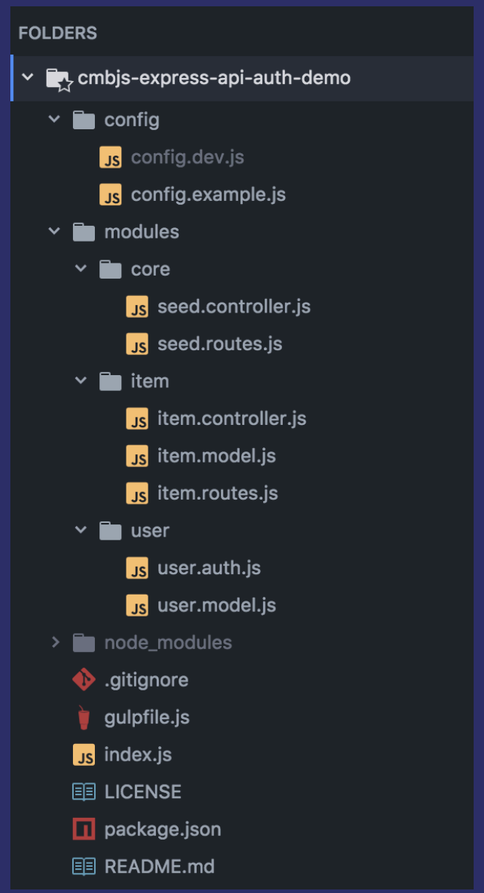
gulp-nodemon

Nodemon is a utility that will monitor for any changes in your source and automatically restart your server.

gulp-livereload

A lightweight gulp plugin for livereload best used with the livereload chrome extension.

Project Organization



Config file

```
//config/config.example.js

'use strict';

//Rename this as config.dev.js with your deatils

module.exports = {
  sessionSecret: process.env.SESSION_SECRET,
  sessionExpiry: process.env.JWT_TOKEN_EXPIRE_TIME || 1200,
  db: {
    uri: process.env.MONGOHQ_URL || process.env.MONGODB_URI ||
 'mongodb://' + (process.env.DB_1_PORT_27017_TCP_ADDR || 'localhost') + '/auth-demo',
    options: {
      useMongoClient: true
    },
    // Enable mongoose debug mode
    debug: process.env.MONGODB_DEBUG || false
  },
  seedDB: {
    seed: process.env.MONGO_SEED === 'true'
  }
};
```

Shall we move to code work though?

Word on validation

```
…

const _ = require('lodash');

const whitelistedFieldsForCreate = [
  'itemCode',
  'itemName',
  'retailPrice'
];
…
module.exports.create = (req, res) => {
  req.body = _.pick(req.body, whitelistedFieldsForCreate);
  let item = new Item(req.body);
  item.save().then((data) => {
    if (!data) {
      res.status(422).json({message: "An error occured saving item!"});
    }
    res.status(200).json(data);
  }).catch((err) => {
    res.status(422).json(err);
  })
};
```

Code will be available on GitHub

https://github.com/iamchathu/cmbjs-express-api-auth-demo

Thank you