

Node Hero – Node.js Authentication using Passport.js

📅 8 months ago

This is the 8th part of the tutorial series called Node Hero – in these chapters, you can learn how to get started with Node.js and deliver software products using it.

In this tutorial, you are going to learn how to implement a local Node.js authentication strategy using Passport.js and Redis.

Upcoming and past chapters:

1. Getting started with Node.js
2. Using NPM
3. Understanding async programming
4. Your first Node.js HTTP server
5. Node.js database tutorial
6. Node.js request module tutorial
7. Node.js project structure tutorial
8. Node.js authentication using Passport.js *[you are reading it now]*
9. Node.js unit testing tutorial
10. Debugging Node.js applications
11. Node.js Security Tutorial
12. How to Deploy Node.js Applications
13. Monitoring Node.js Applications

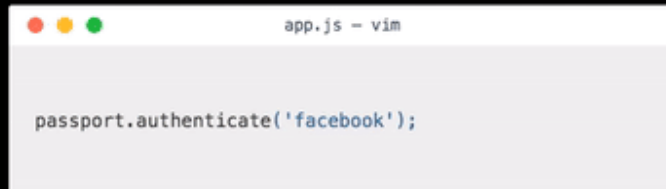
Technologies to use

Before jumping into the actual coding, let's take a look at the new technologies we are going to use in this chapter.

What is Passport.js?

Simple, unobtrusive authentication for Node.js – <http://passportjs.org/>

Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be unobtrusively dropped in to any Express-based web application. A comprehensive set of strategies support authentication using a username and password, Facebook, Twitter, and more.



Passport is an authentication middleware for Node.js which we are going to use for session management.

What is Redis?

Redis is an open source (BSD licensed), in-memory data structure store, used as database, cache and message broker. – <https://redis.io/>

We are going to store our user's session information in Redis, and not in the process's memory. This way our application will be a lot easier to scale.

The Demo Application

For demonstration purposes, let's build an application that does only the following:

- exposes a login form,
- exposes two protected pages:
 - a profile page,
 - secured notes

Node.js Monitoring and Debugging from the Experts of RisingStack

Build performant applications using Trace

START MY FREE TRIAL ►

The Project Structure

You have already learned [how to structure Node.js projects](#) in the previous chapter of Node Hero, so let's use that knowledge!

We are going to use the following structure:

```
├─ app
|   └─ authentication
|   └─ note
|   └─ user
|   └─ index.js
|   └─ layout.hbs
├─ config
|   └─ index.js
├─ index.js
└─ package.json
```

As you can see we will organize files and directories around features. We will have



RisingStack

Never miss an update!

Subscribe

**NODE.JS
DEBUGGING**



Insert the following authentication flow into our application:

97
Shares

1. User enters username and password
2. User Service checks if they are matching
3. If matching, it sends a `Set-Cookie` header that will be used to
4. Access other pages

START MY FREE TRIAL!

4. When the user visits pages from the same domain, the previously set cookie will be added to all the requests
5. Authenticate restricted pages with this cookie

To set up an authentication strategy like this, follow these three steps:

Step 1: Setting up Express

We are going to use Express for the server framework – you can learn more on the topic by reading our [Express tutorial](#).

```
// file:app/index.js
const express = require('express')
const passport = require('passport')
const session = require('express-session')
const RedisStore = require('connect-redis')(session)

const app = express()
app.use(session({
  store: new RedisStore({
    url: config.redisStore.url
  }),
  secret: config.redisStore.secret,
  resave: false,
  saveUninitialized: false
})
})
```



RisingStack

Never miss an update!

your email

Subscribe

**NODE.JS
DEBUGGING**

97
Shares

and all the dependencies that the session management needs. I created a new instance from the `express-session` module, which

we are using Redis, but you can use any other, like MySQL or

START MY FREE TRIAL!

Step 2: Setting up Passport for Node.js

Passport is a great example of a library using plugins. For this tutorial, we are adding the `passport-local` module which enables easy integration of a simple local authentication strategy using usernames and passwords.

For the sake of simplicity, in this example, we are not using a second backing store, but only an in-memory user instance. In real life applications, the `findUser` would look up a user in a database.

```
// file:app/authenticate/init.js
const passport = require('passport')
const LocalStrategy = require('passport-local').Strategy

const user = {
  username: 'test-user',
  password: 'test-password',
  id: 1
}

passport.use(new LocalStrategy(
  function(username, password, done) {
    findUser(username, function (err, user) {
      if (err) {
        return done(err)
      }
    })
  })
)
```



RisingStack

Never miss an update!

your email

Subscribe

NODE.JS DEBUGGING

null, false)



97
Shares

turns with our user object the only thing left is to compare the user's password and the real password to see if there is a match.

START MY FREE TRIAL!

If it is a match, we let the user in (by returning the user to passport – `return done(null, user)`), if not we return an unauthorized error (by returning nothing to passport – `return done(null)`).

Step 3: Adding Protected Endpoints

To add protected endpoints, we are leveraging the middleware pattern Express uses. For that, let's create the authentication middleware first:

```
// file:app/authentication/middleware.js
function authenticationMiddleware () {
  return function (req, res, next) {
    if (req.isAuthenticated()) {
      return next()
    }
    res.redirect('/')
  }
}
```

It only has one role if the user is authenticated (has the right cookies) it simply calls the next middleware; otherwise it redirects to the page where the user can log in.

Using it is as easy as adding a new middleware to the route definition.



RisingStack

Never miss an update!

Subscribe

**NODE.JS
DEBUGGING**

`passport.authenticationMiddleware(), renderProfile)`



Authentication for Node.js with Passport is a piece of cake
RisingStack #nodejs

97
Shares

CLICK TO TWEET

START MY FREE TRIAL!

In this Node.js tutorial, you have learned how to add basic authentication to your application. Later on, you can extend it with different authentication strategies, like Facebook or Twitter. You can find more strategies at <http://passportjs.org/>.

The full, working example is on GitHub, you can take a look here:
<https://github.com/RisingStack/nodehero-authentication>



Get the whole **Node Hero** series as a single pdf and read it later.

Next up

The next chapter of Node Hero will be all about [unit testing Node.js applications](#). You will learn concepts like unit testing, test pyramid, test doubles and a lot more!

Share your questions and feedbacks in the comment section.



RisingStack

Never miss an update!

your email

Subscribe

**NODE.JS
DEBUGGING**



, organizer of @Oneshotbudapest

97
Shares

help with Node.js ?

START MY FREE TRIAL!

the Experts of RisingStack
for your enterprise-grade projects!



HIRE US

nodehero nodejs tutorial authentication security

Read more from us:

- Setting up Node.js on Windows 10
- How to Get Node.js Logging Right

Get early access to our posts

Subscribe



RisingStack

Never miss an update!

Subscribe

NODE.JS DEBUGGING



Stack Blog

Login ▾

Share

Sort by Best ▾

97
Shares

Discussion...

8 months ago

START MY FREE TRIAL!

I suggest using JsonWebTokens instead of cookies, as it makes concern separation (API >< frontend) much easier. If JWT auth is set up properly, developers could also skip the utilization of Redis.

5 ^ | ▾ • Reply • Share ›

**Tracker1** → Poduszló Kristóf • 8 months ago

It depends on your approach.. jwt works for SPA + API access, but for multi-page applications/sites you're still going to want a cookie based implementation for initial server validation/render.

1 ^ | ▾ • Reply • Share ›

**Ming Hann** → Tracker1 • 5 months ago

why multi pages app can't use jwt?

^ | ▾ • Reply • Share ›

**Tracker1** → Ming Hann • 5 months ago

If you're doing a multi page app, you should be doing server side rendering.. how does the browser get a jwt token to the server for first tender on a server? You need a cookie for that... Jwt is kind of big for a cookie and as to each request payload... For multi page, best to use session cookie and look up what you need.

^ | ▾ • Reply • Share ›

**алик** → Tracker1 • a month ago

Is it obligatory for a multipage site to store the session cookies on a serverside?

^ | ▾ • Reply • Share ›

**zoonman** → Poduszló Kristóf • a day ago

RisingStack

Never miss an update!

your email

Subscribe

**NODE.JS
DEBUGGING**

Poduszló Kristóf • a month ago

then to store the user sessions? And how to handle the app without the



START MY FREE TRIAL!

then to store the user sessions? And how to handle the app without the

Reply • Share ›

Poduszło Kristóf • 8 months ago

passport.js supports JWT too :)

Reply • Share ›

Francis Kim • a month ago

s, do support. But for what those field with jwtsession: false?

^ | v • Reply • Share ›



Chemi Martinez • 7 months ago

Hello. Without any hope this is gonna be answered...

I can't get this example working. It seems to be failing when setting the cookie.

In middleware.js i've added some console messages:

```
function authenticationMiddleware() {
  return function (req, res, next) {
    if (req.isAuthenticated()) {
      console.log('auth ok')
      return next()
    }
    console.log('auth error')
    res.redirect('/')
  }
}
```

module.exports = authenticationMiddleware

For whatever reason, it always goes to 'auth error' wether I log in correctly or not.

Please, ¿any help?

2 ^ | v • Reply • Share ›



Gergely Németh → Chemi Martinez • 7 months ago

Did you set the environment variables described in this file:



RisingStack

Never miss an update!

your email

Subscribe

**NODE.JS
DEBUGGING**

or any, thanks for your time.

think I did set them. Im using IntelliJ idea and tried to execute mode

ex is using the env variables but it said 'wrong command' so I ser the



...js using the env variables but it said wrong command so I set the variables directly in my system. First i tries localhost:3000 but it said that it needed slashes //. Then everything ran ok but it didn't work. After that, I changed REDIS_URI to redis://localhost:3000. It tan again but still not working. Any ideas? Thanks!

▼ • Reply • Share ›

97
Shares

Gergely Németh → Chemi Martinez • 7 months ago

REDIS_URI should point to your Redis installation- you can install it from: <http://redis.io>

The default redis port is 6379

1 ^ | ▼ • Reply • Share ›



Chemi Martinez → Gergely Németh • 7 months ago

Hey Gergely. I was typing on a mobile so I couldn't say all I wanted. I wanted to thank you for your insanely fast response.

I installed Redis from this link -> <https://github.com/MSOpenTech/...>

Stable Version 2.8.2400

Now everything works correctly. However, I have 2 questions.

1.- Where do you store the info that says a user is logged in the system? I mean, if I reload the page or type the user and password wrong, and then I go directly to the /profile page it says I'm authenticated and I can see that page.

2.- I guess that if I want more than 1 user I have to use an array or something similar instead of the "const user", and modify the findUser function, right?

Finally, probably I'll have some more questions as I try to understand all the code. May I ask? Thank you again!

^ | ▼ • Reply • Share ›



Gergely Németh → Chemi Martinez • 7 months ago

it is handled by the redis-store module - you can check here: <https://github.com/tj/connect-...>

^ | ▼ • Reply • Share ›



RisingStack

Never miss an update!

your email

Subscribe

**NODE.JS
DEBUGGING**

Gergely Németh → Chemi Martinez • 7 months ago

the secret is for securing the cookies



the secret is for securing the cookie

^ | v • Reply • Share ›

Chemi Martinez → Gergely Németh • 7 months ago

I'll try tonight. I'm not at home right now. Ty!!!

^ | v • Reply • Share ›

97
Shares

n • 8 months ago

Can somebody explain me the blow lines of authentication middleware.

START MY FREE TRIAL! ... ated()) {

```
return next()
}
```

Who is populating in isAuthenticated method ? is it passport ? what happens exactly during that method call ?

Thanks

2 ^ | v • Reply • Share ›



Gergely Németh → mohammed saleem • 8 months ago

Hey,

it just check the cookie - it is added by passport. Does it make sense?

^ | v • Reply • Share ›



joe3487 • 8 months ago

I don't understand the value of using Passport if you have to do all the implementation of making db calls to check username and password. Its just as easy to do this yourself.

What am I missing?

2 ^ | v • Reply • Share ›



letmecode → joe3487 • 8 months ago

I don't think it's as easy as you think it is. Passport is about modularity. It makes it easy to use the set of authentication methods you need. It cannot do database related tasks for you directly without severely restricting itself or violating principles like DRY or "Do one thing and do it well".

Each plugin does exactly one thing, but you're setting the guidelines and details – that's the kind of modularity Express is built upon. You need authentication via Twitter and Facebook? Pull'em in and implement the details. Don't need it anymore?

Then the modules get out and everything else works as it did before.



RisingStack

Never miss an update!

your email

Subscribe

**NODE.JS
DEBUGGING**

authentication call from the client (and how to work in the
cate() call, etc)

• Share ›



2 months ago

you for this very nice tutorial series I really enjoy and appreciate it.

with my question, but looking at the provided code I wonder how you

```
re').init(app)
```

START MY FREE TRIAL!

h nested routes. I tried naively to reuse this pattern inside another "initFeature(app)" functions but with no success. I know of the express.Router but I'm always looking for alternatives.

BR

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Joseph Chambers** • 3 months ago

How do you see the notes? When I goto /note or /notes/ it's telling me Cannot GET /notes

I am logged in as I see:

Hey, admin!

You successfully logged-in \o/

Second note: If I'm creating an admin you would suggest inside the app to have for example a admin/reports or admin/promotions/ and inside promotions I would have the add gift card functionality?

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Debajit Majumder** ➔ Joseph Chambers • 3 months ago

There is no route for notes or note. As per above code you can access a particular note by /notes/:id, e.g. /notes/1

[^](#) | [v](#) • [Reply](#) • [Share](#) ›**Ray C** • 3 months ago

Hi. Thanks for great guide.

However as I m new to Redis, I really do not get the Redis part.

I have installed redis, and tried to run your code, but it seems never saves any session.

Actually I m not sure what is redis stor url , and what should be put as redis-store-url in



RisingStack

Never miss an update!

[Subscribe](#)**NODE.JS
DEBUGGING**[Share](#) ›



Gergely Németh → Ray C • 3 months ago

The url is just the location of the redis instance you want to use to store. By your local machine it will be redis://localhost:6379

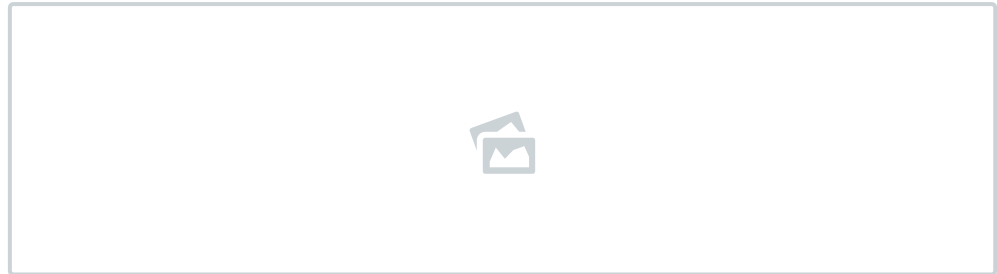
97
Shares

be set using environment variables - it is not a node.js specific thing.
run your app with "node app.js", simply prepend the env vars:
ENV=production node app.js". this way process.env.NODE_ENV will be n."

^ | v • Reply • Share ›



Ray C → Gergely Németh • 3 months ago



Well, sorry but still I can not get it working...
will you advise what caused following error?

^ | v • Reply • Share ›



Ankur Marwaha • 5 months ago

<https://github.com/RisingStack...>

using this

after trying to login, it doesn't do anything

don't matter what i give in as username and password, nothing happens (no login, no error, nothing in the console)

^ | v • Reply • Share ›



Ankur Marwaha • 5 months ago

a general question,

We do "npm install" (to install packages from package.json) before we do "node script.js" (to start the app),

question is where are these packages install (not in control panel), are they installed somewhere internal to nodejs, and if i delete the folder in future, are all the installations also



RisingStack

Never miss an update!

your email

Subscribe

NODE.JS
DEBUGGING

call



dependencies specified in **package.json** are installed into a folder called **node_modules** in the current directory.

Reply • Share ›

Ankur Marwaha → Arjun Mayilvaganan • 5 months ago

Copying and pasting the node_modules folder also work fine ?

▼ • Reply • Share ›

97
Shares

START MY FREE TRIAL!



Arjun Mayilvaganan → Ankur Marwaha • 5 months ago

Yes, it will, as long as it contains all the dependencies.

^ | ▼ • Reply • Share ›



Gergely Németh → Arjun Mayilvaganan • 5 months ago

One gotcha here - if they include native dependencies, it won't work between different Node.js versions. In this case you can still do `npm rebuild`.

But to avoid the headaches I would recommend doing `npm install` for all the projects you do. If you don't have internet connectivity, you can still do `npm install --cache-min 999999` - more info here:

<https://github.com/npm/npm/iss...>

^ | ▼ • Reply • Share ›



Arjun Mayilvaganan → Gergely Németh • 5 months ago

Thanks for that information.

I meant, in cases similar to pushing to a repository without ignoring the `node_modules` folder.

In such cases, should work fine, no?

^ | ▼ • Reply • Share ›



Gergely Németh → Arjun Mayilvaganan • 5 months ago

Not sure if I understood your question - the node modules folder should always be ignored from version control.

^ | ▼ • Reply • Share ›



sevnur • 6 months ago

hi! i couldn't understand that; in which file i am giving the url of the direction that i want to use this authentication??

^ | ▼ • Reply • Share ›



RisingStack

Never miss an update!

your email

Subscribe

NODE.JS
DEBUGGING

on throws an error when I add it and is happy when I remove the



START MY FREE TRIAL!

Share ›

Justin Peake → Justin Peake • 8 months ago

an error elsewhere.

Reply • Share ›

ago

Authentication is requested topic to cover for @risingstack. I have been using it but some good production recipes could be useful.

^ | v • Reply • Share ›



Gergely Németh → ga19892 • 8 months ago

Got it, thanks! We already have a post idea in the pipeline on JWT :)

2 ^ | v • Reply • Share ›



алик → Gergely Németh • a month ago

Where? Any gist, repo sample?

^ | v • Reply • Share ›



daviddavis • 8 months ago

I cloned this project git repo, npm installed and started the app, and upon entering a false username and password combo, or a true combo- the app does nothing after hitting submit, no console log or anything. Am I missing something here? Thanks.

^ | v • Reply • Share ›



Gergely Németh → daviddavis • 8 months ago

Hello David,

do you have a redis up and running? Make sure to set these env vars:

<https://github.com/RisingStack...>

^ | v • Reply • Share ›



daviddavis → Gergely Németh • 8 months ago

Oh, that must be it. Im unsure how to set redis up properly. I gave it a shot in the dark but it's not working. Here's what I did and it's not working:

```
const config = {}
```



RisingStack

Never miss an update!

your email

Subscribe

NODE.JS DEBUGGING

secret: process.env.REDIS_STORE_SECRET

};



START MY FREE TRIAL!

module.exports = config;

• Reply • Share ›

Gergely Németh → dauiddavis • 8 months ago

Hey,

those are environment variables, which you can set in the command line using the following command:

REDIS_STORE_URI=localhost:5672

REDIS_STORE_SECRET=awesome node index.js

^ | v • Reply • Share ›



dauiddavis → Gergely Németh • 8 months ago

Ok got it. Thanks.

^ | v • Reply • Share ›



BenBoy CodeHeroes • 8 months ago

Great article! I love reading the blog.

There's an issue with the link to "Node.js project structure tutorial". It redirects me back to the current page.

^ | v • Reply • Share ›



Gergely Németh → BenBoy CodeHeroes • 8 months ago

We are happy that you like it!

Thanks, just fixed it!

^ | v • Reply • Share ›



Subscribe



Add Disqus to your site

Add Disqus Add



Privacy



RisingStack

Never miss an update!

your email

Subscribe

**NODE.JS
DEBUGGING**

97
Shares



97
Shares

START MY FREE TRIAL!

Blog

- Best articles
- Node Hero Tutorials
- Node.js at Scale Tutorials

Resources

- Node.js Maturity Checklist
- Trace by RisingStack – Node.js Monitoring
- Node.js is Enterprise Ready
- RisingStack Community
- Node.js Daily

Connect

- RisingStack.com
- Twitter
- Github
- Facebook



Never miss an update!

your email

Subscribe

