

# Node Auth Tutorial (JWT)

## #15 - Checking the Current User



```
const jwt = require('jsonwebtoken');
const User = require('../models/User');

const requireAuth = (req, res, next) => {
  ...
};

// check current user
const checkUser = (req, res, next) => {
  const token = req.cookies.jwt;
  if (token) {
    jwt.verify(token, 'secret key', async (err, decodedToken) => {
      if (err) {
        res.locals.user = null;
        next();
      } else {
        let user = await User.findById(decodedToken.id);
        res.locals.user = user;
        next();
      }
    });
  } else {
    res.locals.user = null;
    next();
  }
};

module.exports = { requireAuth, checkUser };
```

**Insert code**

A diagram with the text 'Insert code' on the right. Two arrows originate from it: one points to the first code block (the imports) and the other points to the second code block (the middleware functions).

```
const express = require('express');
const mongoose = require('mongoose');
const authRoutes = require('./routes/authRoutes');
const cookieParser = require('cookie-parser');
const { requireAuth, checkUser } = require('./middleware/authMiddleware');
```

...

// routes

```
app.get('*', checkUser);
app.get('/', (req, res) => res.render('home'));
app.get('/smoothies', requireAuth, (req, res) => res.render('smoothies'));
app.use(authRoutes);
```

...

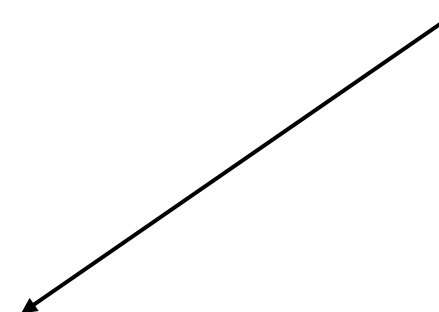
**Insert code**

Two arrows originate from the text 'Insert code'. The top arrow points to the 'checkUser' property in the destructured import statement of the first code block. The bottom arrow points to the 'checkUser' argument in the first route handler of the second code block.

```
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="/styles.css">
</head>

<body>
  <nav>
    <h1><a href="/">Master Chef Smoothies</a></h1>
    <ul>
      <% if (user) { %>
        <li>Welcome, <%= user.email %></li>
        <li><a href="/logout">Log out</a></li>
      <% } else { %>
        <li><a href="/login">Log in</a></li>
        <li><a href="/signup" class="btn">Sign up</a></li>
      <% } %>
    </ul>
  </nav>
```

**Insert code**

Master Chef  
Smoothies

Welcome,  
mario@gmail.com

LOG  
OUT



# Smoothie Recipes

By Master Chef For Your

[VIEW RECIPES](#)

Copyright 2021 Master Chef Smoothies

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
  - http://localhost:3000
- Trust Tokens

Cache

- Cache Storage
- Application Cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging

Frames

- top

Filter

Name	Value	Domain	Path	Expires / M...	Size	HttpOnly	Se...	S...	S...	P...
jwt	eyJhbGciOiJI...	localhost	/	2021-09-03...	174	✓				M...
isEmployee	true	localhost	/	2021-08-31...	14	✓				M...

Cookie Value

☐ Show URL decoded

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYxMmRkNjIIMzgzNGZmNzBiYmUwMjdmYSIsImh0dCI6MTYzMDQxNDUxMCwiZiXhwldjoxNjMwNjczNzEwfQ.plKz6ztGAyjNnpwL6A5\_ZR9g1K0Uw0c0q8bRUUp0\_gHI

Master Chef Smoothies

LOG IN

SIGN UP



# Smoothie Recipes

By Master Chef For Your

[VIEW RECIPES](#)

Copyright 2021 Master Chef Smoothies

Elements Console Sources Network Performance Memory Application Security Lighthouse

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
  - http://localhost:3000
- Trust Tokens

Cache

- Cache Storage
- Application Cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync
- Push Messaging

Frames

- top

Filter

- ☐ Only show cookies with an issue

Name	Value	Domain	Path	Expires / M...	Size	HttpOnly	Se...
isEmployee	true	localhost	/	2021-08-31...	14	✓	

Select a cookie to preview its value