

Authentication



Daniel Stern

CODE WHISPERER

<http://github.com/danielstern>



“Love all, trust a few.”

William Shakespeare - *All's Well That Ends Well*



What Is Authentication?



An individual claims to be a certain user and demands access to certain data



Authentication determines if the individual is the user in question

Authentication Overview



Process that determines if user has permission to access data



User password can be compared to hash on DB



Allows multiple users to have separate, private data



Necessary for most e-commerce activities



Authentication: More Than Passwords

Password



Fingerprint, Retinal Scan, Microchip



Third-party confirmation (Facebook, Twitter)



State-issued documents



Authentication Roadmap



Create route guard that reroutes unauthenticated users to new login page

Create login page with form that communicates with server

Update default state to include authentication information

Create authentication route on server which grants access only to users with valid login information



Security Reminder:

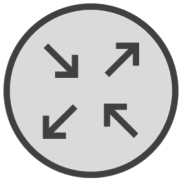
Complete at least a few web security courses before attempting to secure your application for use in the real world.



Adding Route Guards



What Are Route Guards?



Some routes should only be accessed by users with correct credentials



Route guards organize authentication logic at router level

403
Forbidden

Typically, Route Guards redirect to a “Login” or “403” page when credentials are not available



Demo



Add route guard to every existing route

- Checks if state is loaded (could also check for expired login info, etc.)
- Redirects user to login page if there is no state

Create login page stub for redirection



Creating a Login Page and Login Saga



Demo



Update default state to include passwords and reinitialize DB

Create login page with form that allows user to input name and password

Create a login saga which confirms login info with the server



Adding Authentication to the Server



Demo



Add authentication route which validates user credentials and returns appropriate response

Add utility to server that assembles user state on-demand upon successful authentication



Finalizing Integration between Client and Server



Demo



Create mutation that applies the application state returned by the authentication response to the Redux store

Verify that our full stack application is functional!



Authentication Summary



Restricting access to data and routes is an elementary feature of full stack applications

Express routes can be used to validate user-provided authentication data

Route guards can be used to organize authentication logic on client



Coming up in the Next Module...



Configuring Heroku

Updating URLs for deployments

Deploying the application to production

