

Your Next Week

Saturday April 25

6:30 PM
— DUE Class 12 Code Challenge

— DUE Class 12 Lab
— DUE Class 13 Reading
— Class 13

MIDNIGHT
— DUE Class 13 Learning Journal

Sunday April 26

MIDNIGHT
— DUE Class 12-13 Feedback

Monday April 27

6:30 PM
— Career Coaching Workshop #1 Continued (Mandatory)

Tuesday April 28

6:30 PM
— DUE Class 13 Lab
— DUE Class 13 Code Challenge
— DUE Class 14 Reading
— Class 14A

Wednesday April 29

6:30 PM
— Class 14B

MIDNIGHT
— DUE Class 14 Learning Journal

Thursday April 30

6:30 PM
— Co-working

Friday May 1

Saturday May 2

6:30 PM
— DUE Class 14 Mock Interviews
— DUE Class 14 Lab
— DUE Class 15 Reading
— Class 15
— Interview Prep 01

MIDNIGHT
— DUE Class 15 Learning Journal

What We've Covered

<p><i>Module 01</i> Javascript Fundamentals and Data Models</p> <p>C01 — <i>Node Ecosystem, TDD, CI/CD</i> C02 — <i>Classes, Inheritance, Functional Programming</i> C03 — <i>Data Modeling & NoSQL Databases</i> C04 — <i>Advanced Mongo/Mongoose</i> C05 — <i>DSA: Linked Lists</i></p>	<p><i>Module 02</i> API Servers</p> <p>C06 — <i>HTTP and REST</i> C07 — <i>Express</i> C08 — <i>Express Routing & Connected API</i> C09 — <i>API Server</i> C11 — <i>DSA: Stacks and Queues</i></p>	<p><i>Module 03</i> Auth/Auth</p> <p>C10 — <i>Authentication</i> C12 — <i>OAuth</i> C13 — Bearer Authorization C14 — Access Control (ACL) C15 — DSA: Trees</p>	<p><i>Module 04</i> Realtime</p> <p>C16 — Event Driven Applications C17 — TCP Server C18 — Socket.io C19 — Message Queues C20 — Midterms Prep</p> <p>Midterms</p>
<p><i>Module 05</i> React Basics</p> <p>C21 — Component Based UI C22 — React Testing and Deployment C23 — Props and State C24 — Routing and Component Composition C25 — DSA: Sorting and HashTables</p>	<p><i>Module 06</i> Advanced React</p> <p>C26 — Hooks API C27 — Custom Hooks C28 — Context API C29 — Application State with Redux C30 — DSA: Graphs</p>	<p><i>Module 07</i> Redux State Management</p> <p>C31 — Combined Reducers C32 — Asynchronous Actions C33 — Additional Topics C34 — React Native C35 — DSA: Review</p>	<p><i>Module 08</i> UI Frameworks</p> <p>C36 — Gatsby and Next C37 — JavaScript Frameworks C38 — Finals Prep</p> <p>Finals</p>

Interview Prep

- Two slots, (almost) every Saturday starting today
- 3:30pm, 4:30pm
- Optional attendance, 10 points extra credit (in Lab category) for interviewee
- 24 slots total - [See schedule here](#)
 - Everyone can sign up once
 - Depending on extra slots, we'll either do **bonus lectures** or allow second sign up

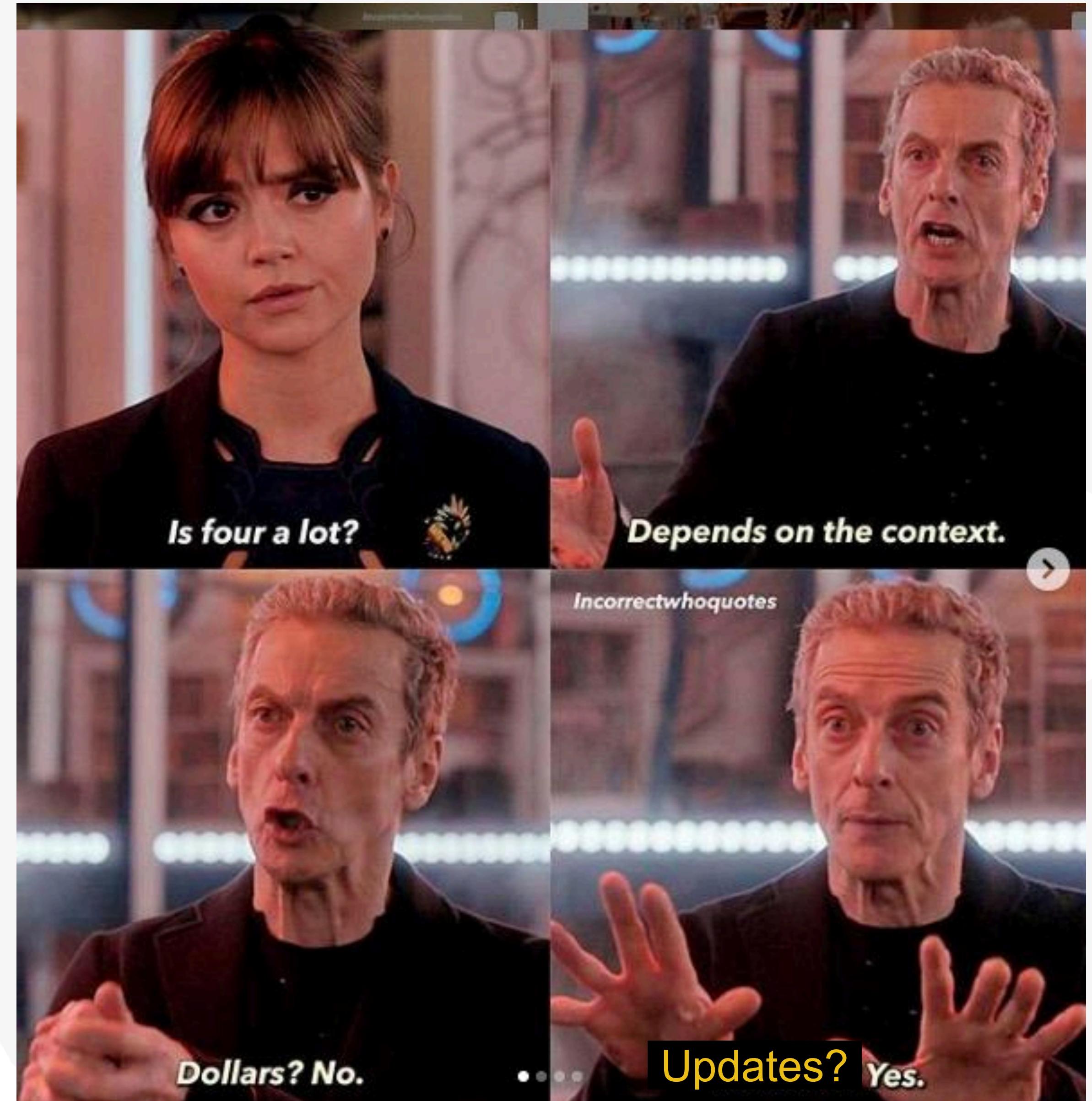
Interviewer: So why do you want this job?

Me: Well, I've always felt passionately about not starving to death



Other Updates

- Thank you all for some great syncs!
- Common thoughts
 - Quarantine sucks
 - Want a better feeling of community
 - Need interview prep/practice
 - Difficult to ask questions
- New weekly slack checkins on Saturday, starting NEXT Saturday
- Wednesday Social Thread



Lab 12 Review

Class 13

Bearer Authorization

seattle-javascript-401n16

JSON Web Token

- A very common way to keep a client logged in
- The server generates a unique token
 - This is an encrypted string
 - Only the server has the encryption key
 - The data encrypted is typically some unique reference to the current user (user id for example)
- **JSON Web Token** is a package that generates tokens for us



NORDICAPIS.COM

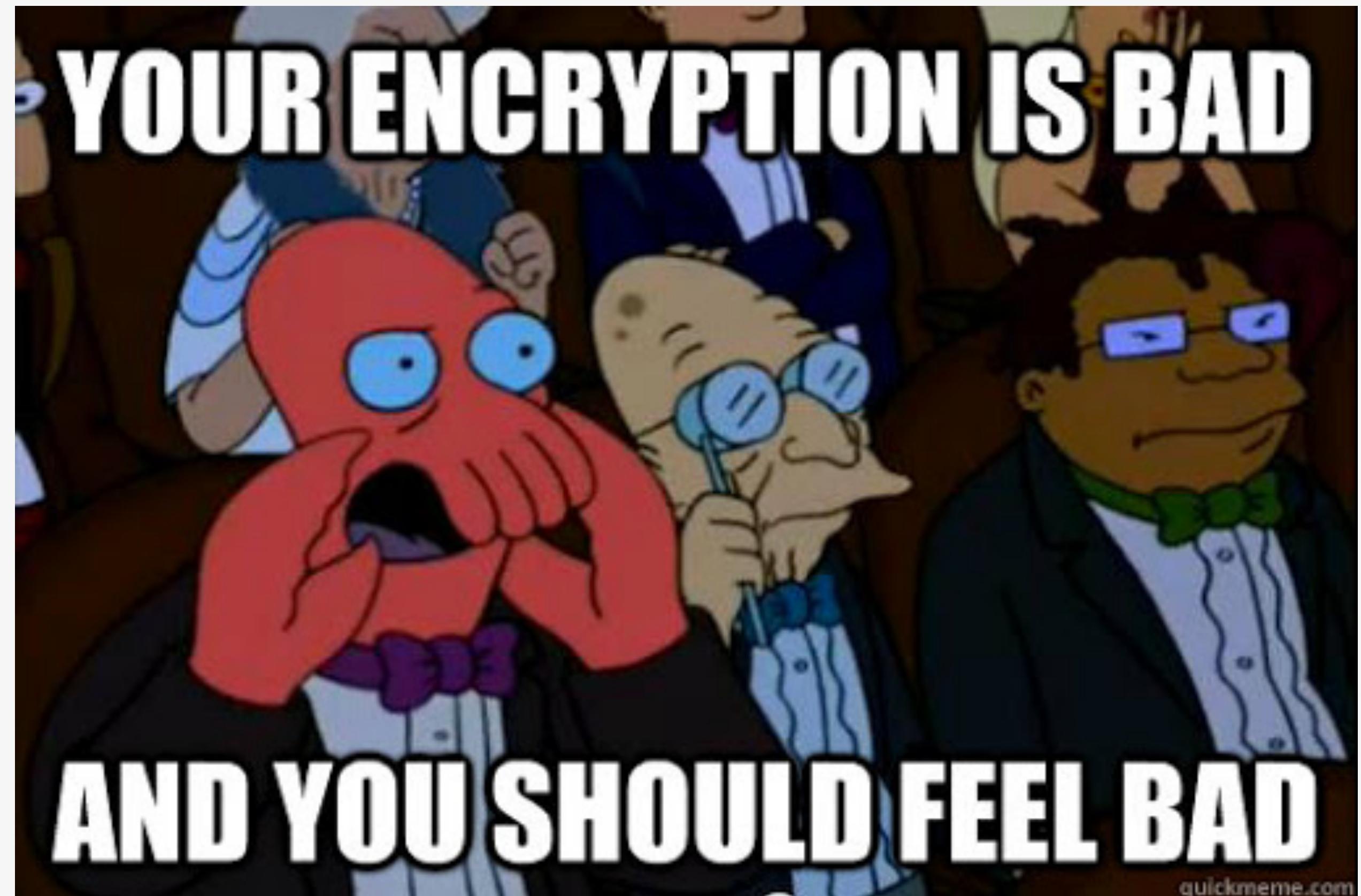
Adding “Bearer”

- When we create a token, we should add the word “Bearer”
- This tells us how to decrypt it later!
- Allows for one auth middleware that handles *everything*
- Client grabs this from response and sends it back on future requests
- We can now hide pages if a Bearer token isn’t present (or redirect to sign in!)



Securing JWT Further

- JSON Web Tokens are “light encryption”
 - Still pretty secure
 - But a little susceptible to hacking
- We can add security by
 - Creating a **timeout** for the JWT
 - Generating a new JWT after every successful client request

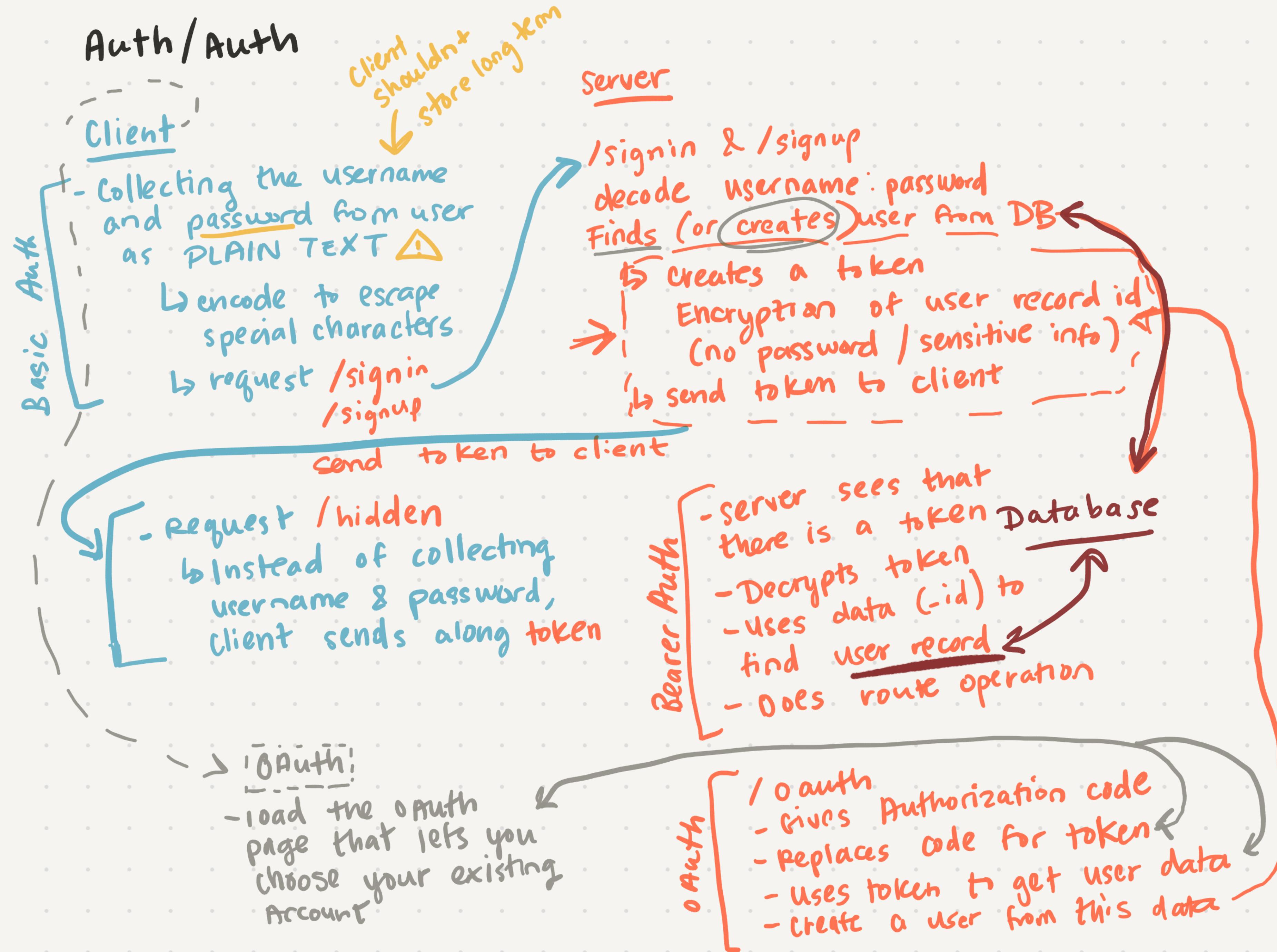


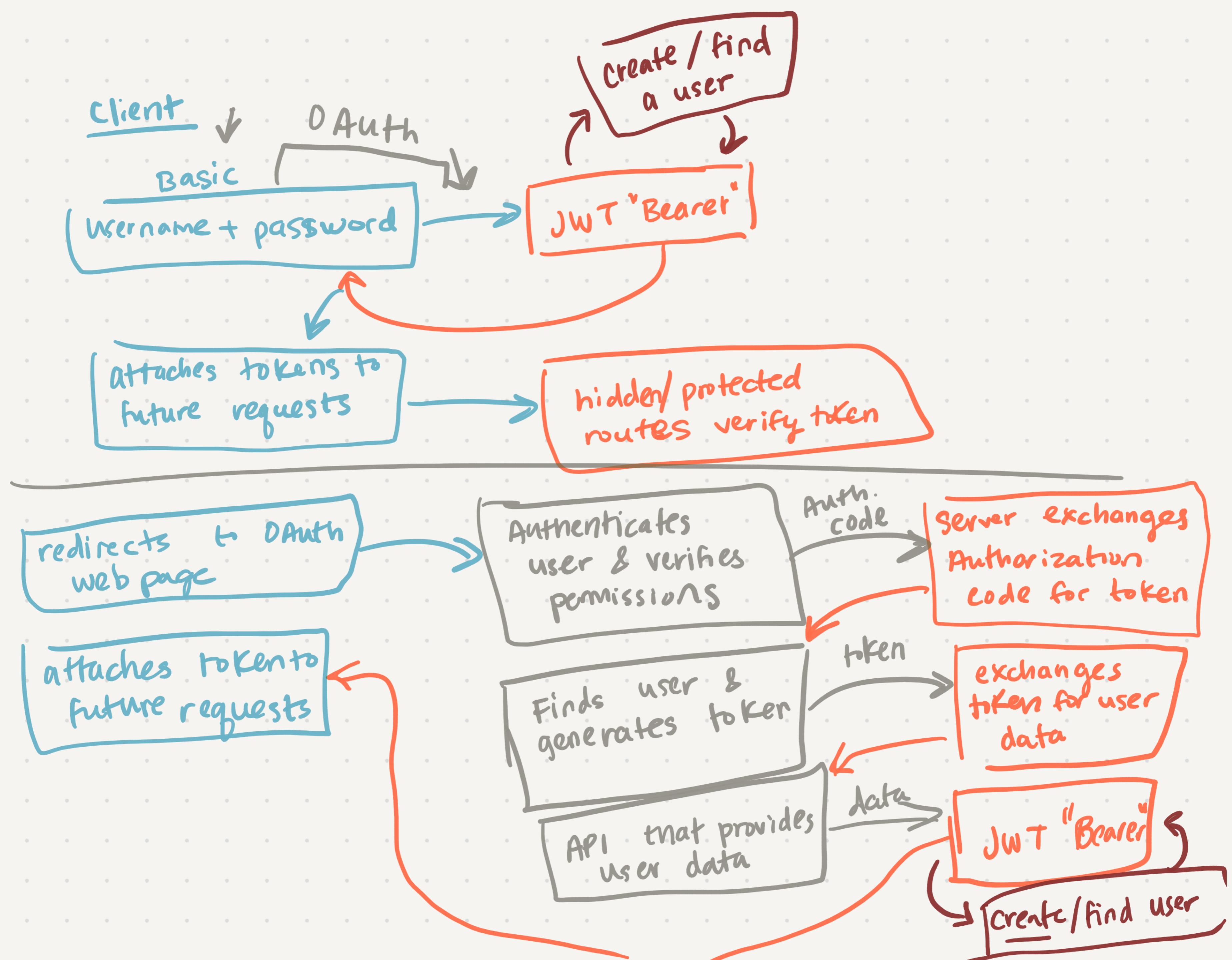
Mongoose Methods

- You can write **methods** upon an individual record
 - Very useful when you want to use a record's information to output something
- You can write **statics** upon an entire model
 - Not so useful; our model class wrapper can handle these operations



Auth / Auth





Lab 13 Overview