

Your Next Week

<p>Saturday April 11</p> <p><i>9 AM</i></p> <ul style="list-style-type: none">— DUE Class 08 Code Challenge— DUE Class 08 Lab— DUE Class 09 Reading— Class 09 <p><i>MIDNIGHT</i></p> <ul style="list-style-type: none">— DUE Class 09 Learning Journal	<p>Sunday April 12</p> <p><i>MIDNIGHT</i></p> <ul style="list-style-type: none">— DUE Career: Professional and Personal Development— DUE Class 08-09 Feedback	<p>Monday April 13</p> <p><i>6:30 PM</i></p> <ul style="list-style-type: none">— Co-working (Optional)	<p>Tuesday April 14</p> <p><i>6:30 PM</i></p> <ul style="list-style-type: none">— DUE Class 09 Mock Interviews— DUE Class 09 Lab— DUE Class 10 Reading— Class 10A
<p>Wednesday April 15</p> <p><i>6:30 PM</i></p> <ul style="list-style-type: none">— Class 10B <p><i>MIDNIGHT</i></p> <ul style="list-style-type: none">— DUE Class 10 Learning Journal	<p>Thursday April 16</p> <p><i>6:30 PM</i></p> <ul style="list-style-type: none">— Co-working	<p>Friday April 17</p>	<p>Saturday April 18</p> <p><i>9 AM</i></p> <ul style="list-style-type: none">— DUE Class 10 Code Challenge— DUE Class 10 Lab— DUE Class 11 Reading— Class 11— Instructor Syncs <p><i>MIDNIGHT</i></p> <ul style="list-style-type: none">— DUE Class 11 Learning Journal

Lab 08 Review

Code Challenge 08

Review

Class 09

API Server

seattle-javascript-401n16

Params

- We've used params to get data from the path
- Anything with a colon = variable to store
- We can also add middleware for routes where a specific param exists!
- Opens the door for generalizing

**CHANGE GOOGLE SEARCH
URL PARAMETERS**



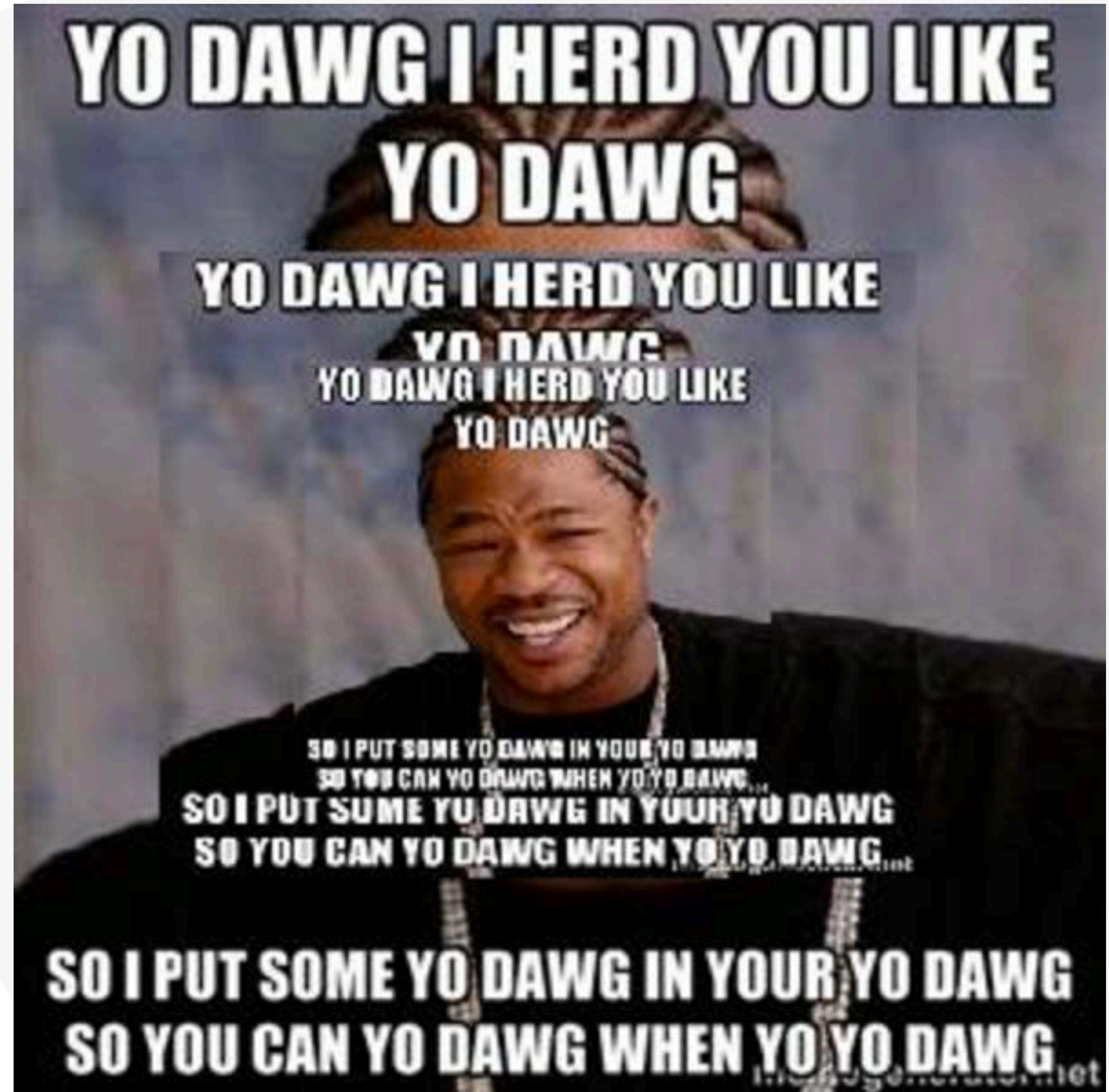
Mongoose Documents

- Remember - a document is a record
- MongoDB is *non-relational*
 - Doesn't enforce any relationship between collections
 - Mongoose can “fake” this



Sub Documents

- A document within a document!
- Basically, a “relationship” is when
 - A record needs to reference another record in another collection
 - When loading a record, we also want to load another record in another collection



Reference Relation

- Most of the time, we create a “relationship” by using just IDs
 - Product belongs to a category
 - Product has a category field
 - This field is set to an ID of a category record
- When you read a product, you can then read the category with the given ID



Virtual Join

- Comes from SQL
- The idea of “joining two tables”
- Basically, creating a link between records of different collections
- Mongoose fakes this by doing another “find” operation
- Mashs the results together
- `populate()` is how we trigger it



Virtual Property

- **Virtual** = not real
- A field that is not “actually” stored in that record
 - Instead the field is generated using a **getter** and **setter**
 - We define what code to run when someone tries to “get”
 - We define what real properties to change when someone tries to “set”



Lab 09 Overview

Code Challenge 09

Overview