Week 2: Data Functions

Review

Last week we come up with the data structure shown to the right, as well as how users would be able to interact with each of the collections.

Here is the list of interactions we came up with for trips:

- create a trip, view all trips for a user, view a trip, update a trip, and delete a trip

(we're just gonna go over trips, cause this covers all of the types of things you'll essentially want to do)

Foreword

MongoDB, as well as Node, has a lot of dumb setup. I'm going to focus on *the important parts* so that you can actually get to coding your project, and we'll talk about putting it all together when it becomes relevant. You're also going to essentially be starting from a template, so you'll have a lot more foundation to go off of.

Get/View/Read functions

Our goal is to read a single trip with a specified id

1. “With a specified id” means our function *must* take in an id to search for
2. First we get the collection of trips
3. Then we search the collection for a document with a matching id
4. Finally we return that trip object

*Note: sometimes functions have **await** before them. This means that the function is asynchronous, but all you really need to know is that **await** should be before all of your mongo functions.

Get/View/Read all functions

Our goal is to read all of the trips for a specified user

1. “For a specified user” means our function *must* take in a user to search for
2. Once we have it, we can get the list of trip *id*s
3. We must then search the trips collection for each of the trips
4. Finally we return a list of trip objects

Update/Patch functions

Our goal is to update a trip with a specified id

1. “With a specified id” means our function *must* take in an id to search for
2. For ease, our functional is also going to take in an object with optional fields
    a. If you want to update the number of flights, then include it in the object, otherwise don’t
3. Get the object currently in the database
4. Update the object locally
5. Update the object stored in the database

Create functions

Our goal is to create a new trip for a specified user

1. “For a specified user” means our function *must* take in a user to add the trip to
    a. What else needs to be included? What doesn’t need to be included?
        i. Example: things like “destination” can’t really have a default value
        ii. Things like “flights” can automatically be set to 0, and added/updated later
2. Create a trip object
3. Add this to the trips collection
4. Find the specified user
5. Add the id of the created trip to the user

Delete/Remove functions

Our goal is to delete a trip with a specified id

1. “With a specified id” means our function *must* take in an id to search for
2. Find and delete the specified trip from the trips collection
3. A user has this trip id in their list of trips, so find that user and delete the trip from their array
4. Technically, you would also want to go through any activities and delete *them*, but we won’t get into that

Mongo function summary

<selector> = { <field>: <value> } => can check equality, or a lot more, more here

<operation> = $set, $push, $pull, more here

findOne(<selector>) => returns the first object that matches the selector

updateOne(<selector>, { <operation>: <value> }) => returns an object with a matchedCount, which will be 0 or 1. If it was 1, the operation succeeded.

insertOne(<value>) => returns an object with a insertedCount, which will be 0 or 1. If it was 1, the operation succeeded.

deleteOne(<selector>) => returns an object with a deletedCount, which will be 0 or 1. If it was 1, the operation succeeded.