Intro To Mongoose

The Coding Bootcamp

Class Objectives

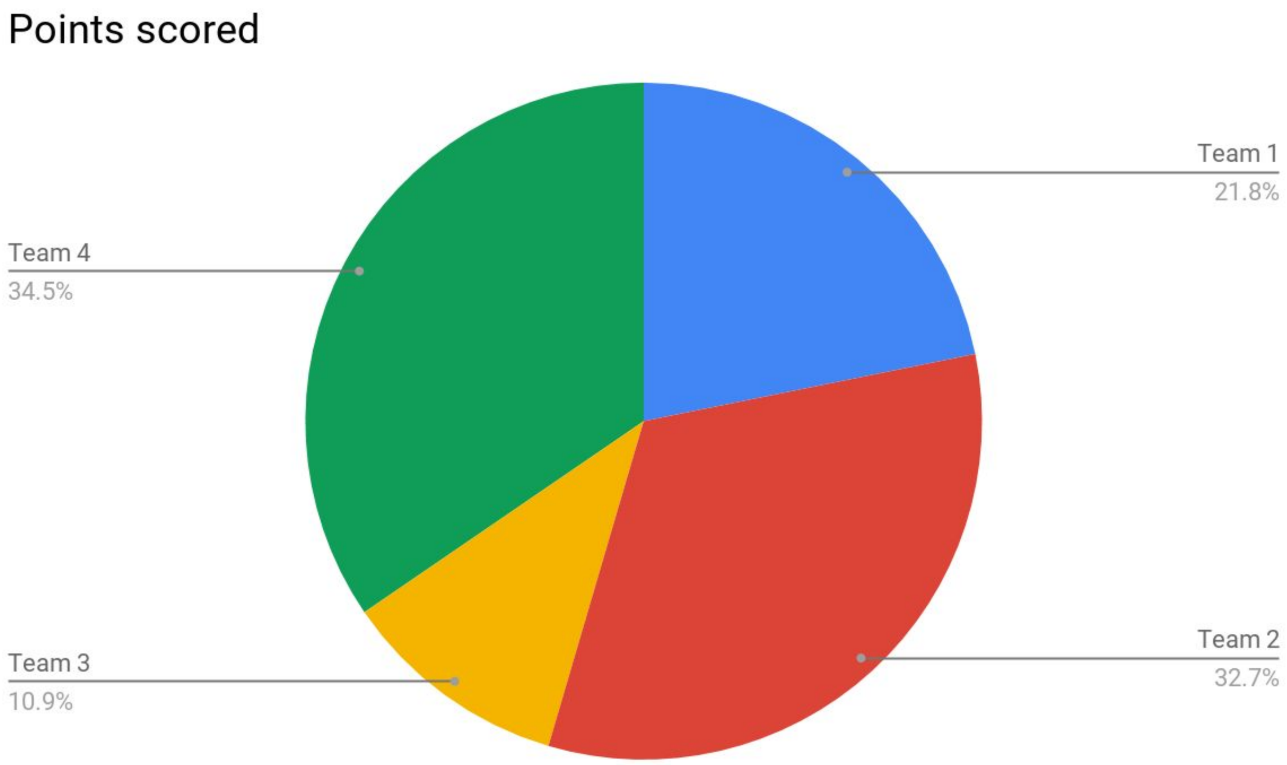
- Understand how to write up a **Mongoose** schema to dictate rules for their **MongoDB** data.
- Create custom methods in Mongoose to set and update data purely on the back end.
- Mongoose's “populate” method to create relationships between the collections in their database.
- Understanding why MySQL may still be more suitable for more complex projects.

Today's Busy Schedule

- Reintroduce Career Services
- MongoJS Refresh (me)
- Warm-Up (us)
- Mongoose (ODM) Intro (me)
- Demo Solved Mongoose Schema (me)
- User Schema (you)
- BREAK
- Custom Methods (me)
- Custom Method Exercise (you)
- Intro Populate (me)
- Populate Exercise (you)

NO, I'M MR. CROWBAR, AND THIS IS MY FRIEND, WHO IS ALSO A CROWBAR.

Project 2 Results…... Monday



Create with MongoJS

```
newAnimal = {“name”:”cat”};

db.animals.save(newAnimal, function(error, saved) {
    // Show any errors
    if (error) {
      console.log(error);
    }
    else {
      // Otherwise, send the response to the client (for
AJAX success function)
      // res.send(saved);
    }
  });
```

Read/Find with MongoJS

```
db.animals.find({“animal”: “cat”}, function(error, found) {
// Throw any errors to the console
   if (error) {
     console.log(error);
   }
   // If there are no errors
      else {
      // Otherwise, send the response to the client (for
AJAX success function)
      //res.json(found);

   }
});
```

Update with MongoJS

```
db.animals.update(
    {
      _id: mongojs.ObjectId(req.params.id)
    },
    {
      // Set “name" to “cat” for the animal we specified
      $set: {
        “name”: “cat”
      }
    },
    // When that's done, run this function
    function(error, edited) {
      // Show any errors
      if (error) {
        console.log(error);
        res.send(error);
      }
      else {
        // Otherwise, send the result of our update to the browser
        //console.log(edited);
        //res.send(edited);
      }
    }
  );
```

09 Stu-MongoJS Review

Instructions

Your goal is to complete the CRUD routes in the server file so the site can display and edit the book data.

Use 07-Stu-Mongo-CRUD_Example or
[Mongo guides](https://docs.mongodb.com/guides/) if you are stuck.

Don't worry about the front end, just use MongoJS to finish the routes

Introducing…

mongoose

elegant mongodb object modeling for node.js

Define models for our Mongo data

Required Fields

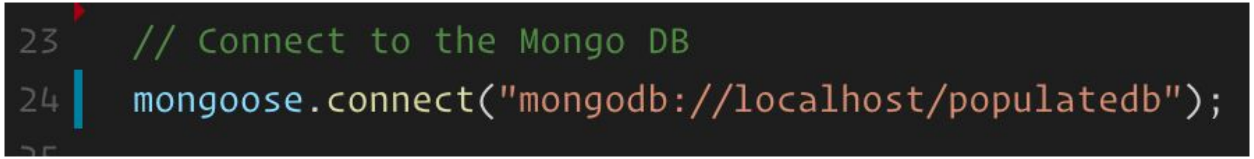
Custom Methods

Combine Collections

Mongoose Models

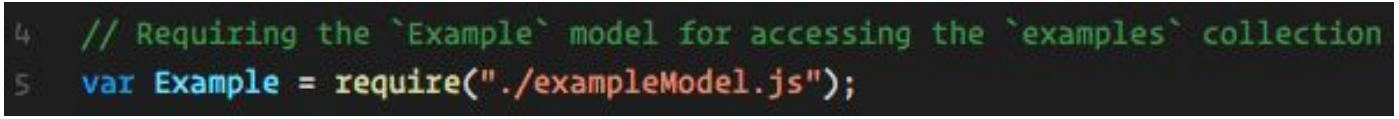
Demo 10-Ins

Uses the url path of the MongoDB database to connect with MongoDB



```
// Connect to the Mongo DB
mongoose.connect("mongodb://localhost/populatedb");
```

Define a schema for the model and then use the model to query our database.



```
// Requiring the `Example` model for accessing the `examples` collection
var Example = require("./exampleModel.js");
```

- **Mongoose models are similar to those in sequelize.**
- We define a schema for the model and then use the model to query our database.

Students: Make a Model Schema (20 mins)

[15-User-Schema]

* Open the `userModel.js` file and complete the User Schema based on the specifications outlined in the file.

* **The only file you will need to modify is `userModel.js`!**

username: A string that will be be required, and also trimmed.
password: A string that will be required, trimmed, and at least 6 characters.
email: A string that must be a valid email address and unique in our collection.
userCreated: A date that will default to the current date.

Hint(s) * The regex for checking if a string is an email is: /.+\@.+\..+/

Custom Methods with Schema (16)

What about when we want more functionality than what these schema properties offer?

```
UserSchema.methods.fullname = function() {
  this.fullname = this.firstname + " " + this.lastname;
  return this.fullname;
};
```

Mongoose provides a way for us to create custom methods to manipulate our data.

Students Do:Custom Method Exercise (10 mins)

[13-Custom-Method-Exercise]

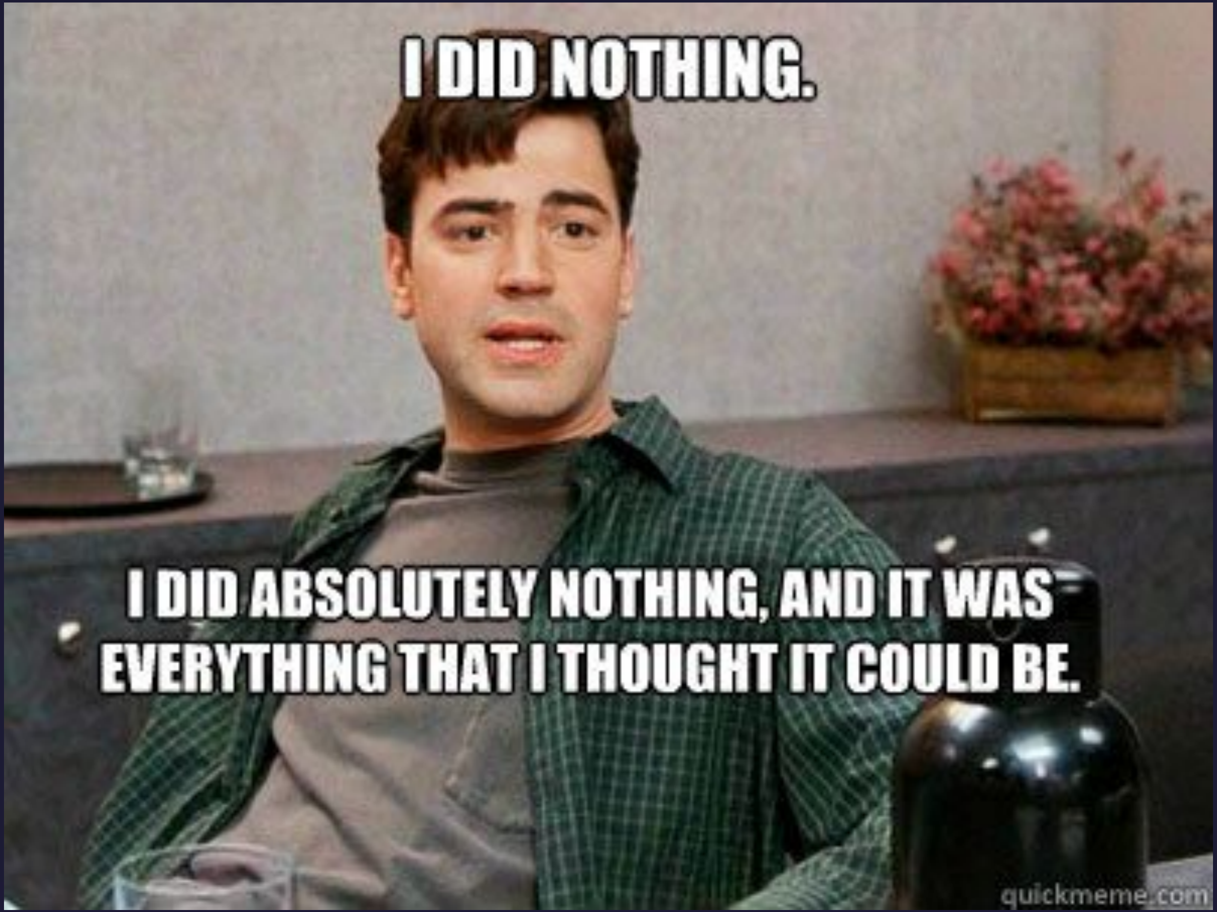
Instructions

1. Go to userModel.js, and create custom methods
  based on the details offered in the file.

2. Once you've made those custom methods, use them
  in this file's POST request

3. Good luck!

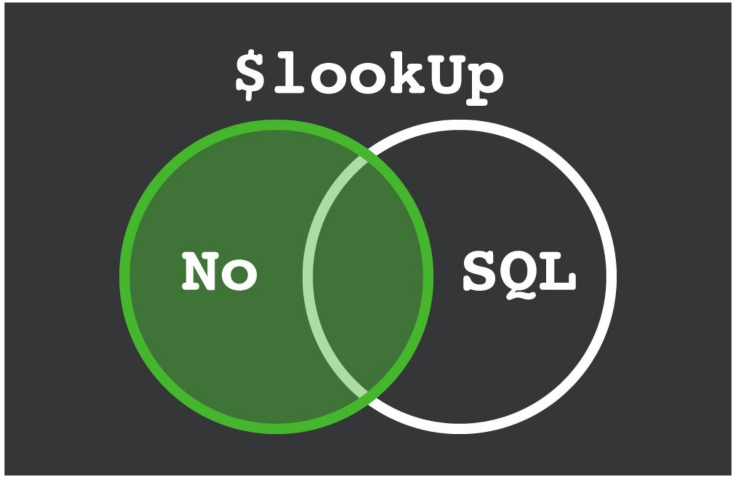
Break (20 mins)


I DID NOTHING.
I DID ABSOLUTELY NOTHING, AND IT WAS EVERYTHING THAT I THOUGHT IT COULD BE.
quickmeme.com

Mongo Population (Library and Books) (14)

Populate lets users relate one collection to another.

```
db.Book
   .find({title:'Coraline'})
     .populate('author')
```



Partners Do: Finish User Model (10 mins)

Instructions

[server.js]

* Open `server.js` and update the `/populate` route to return Users populated with notes as JSON to the client.

Students Do: Scraping with Mongoose

Instructions

* Open `server.js` and complete the empty routes for accessing all articles, accessing a specific article, and for saving a new article.

Questions?