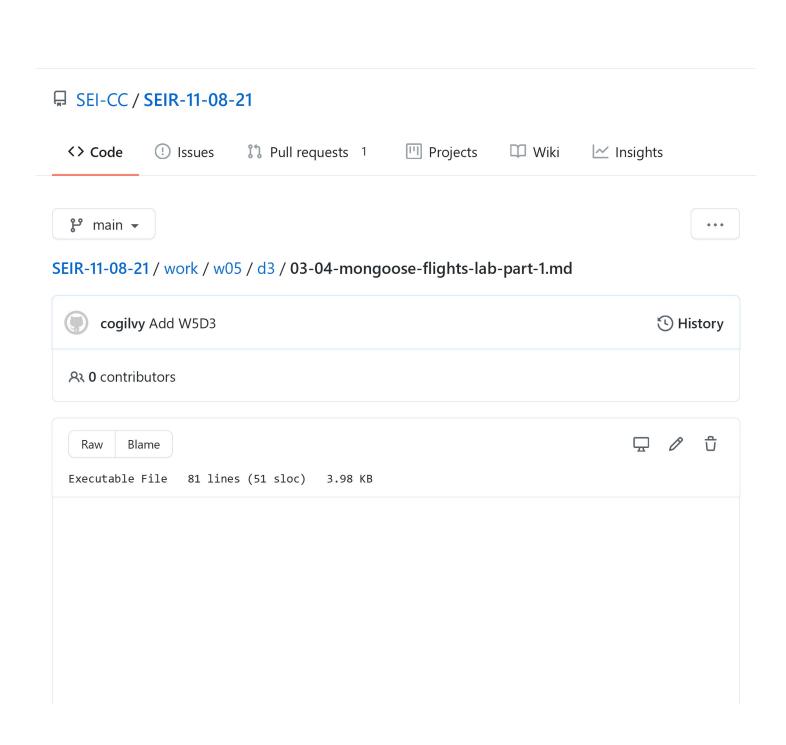


Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide





Mongoose "Flights" Lab - Part 1

Intro

Today in the Intro to Mongoose lesson you Created and Read documents using a Movie Model.

In this lab, you'll do the same, except you'll create and use a $\ensuremath{\mathtt{Flight}}$ model.

Similar to what we did in the lesson, you'll start by creating a mongoose-flights project.

FYI, future lessons will expand upon the mongoose-movies project, and the labs will expand upon the mongoose-flights project!

The final version of mongoose-flights (parts 1 thru 3 combined) will be a deliverable, so do each part and don't fall behind.

Exercises

- 1. Use express generator to create a mongoose-flights project. Be sure to install the node modules after you cd into the project.
- 2. Create a **config/database.js** module that connects to a database named flights . Be sure to require the module in **server.js**.
- 3. Create a Flight Model with the following properties:

Property	Туре	Validations	Default Value
airline	String	enum to include 'American', 'Southwest' & 'United'	n/a
airport	String	enum to include 'AUS', 'DFW', 'DEN', 'LAX' & 'SAN'	'DEN'
flightNo	Number	Required Between 10 and 9999	n/a
departs	Date	n/a	One year from date created

- 4. Implement the following User Stories:
 - AAU, I want to view a list of all flights (index view) that displays each flight's airline, airport, flight no., and departure date/time.
 - AAU, I want to create flights by entering the information on a page (new view) that has a form and submitting it.
 - AAU, I want to be able to access each view via a navigation bar at the top of the page with links to:
 - ALL FLIGHTS, and
 - ADD FLIGHT

Hints:

• Checkout the <input type="datetime-local"> to assist users in entering valid date/time values.

• In the form for adding a new Flight, use a <select name="airport"> to assign the flight's airport . Since they don't change, it's okay to hard-code the <option> elements, e.g., <option value="DEN" selected> .

Bonuses

1. Display the default departure date when displaying the new flight form.

Hints:

i. In the flight controller's new action, you could create an in-memory flight like this:

```
const newFlight = new Flight();
```

This in-memory flight doc would have the default departure date set properly based on the logic in the function you assigned to <code>default</code>.

- ii. Just like any other data you want to access/display in a view template, that data needs to be passed by the controller action when it calls <code>res.render</code>, however...
- iii. Although an input of type="datetime-local" will display a date assigned to its value attribute, that date value needs to be formatted as a string matching this format: yyyy-MM-ddThh:mm (yes, a "T" character is used to separate the date portion from the time portion). One way of obtaining the properly formatted string is to use the following approach:

```
const newFlight = new Flight();
// Obtain the default date
const dt = newFlight.departs;
// Format the date for the value attribute of the input
let departsDate = `${dt.getFullYear()}-${(dt.getMonth() + 1).toString().padStart(
departsDate += `-${dt.getDate().toString().padStart(2, '0')}T${dt.toTimeString().
res.render('flights/new', { departsDate });
```

- 2. Code these additional User Stories:
 - AAU, I want to view the list of flights by departure date in ascending order.
 - AAU, I want the flights in the list to be displayed using red text if the flight's departure date has passed.
- 3. Style the index and new views.

Deliverable?

The final version of mongoose-flights (parts 1 thru 3 combined) will be a deliverable - don't fall behind.

Please create a repo named mongoose-flights and commit your code.

The link to the repo will be submitted using the form as explained in the class readme.