Learn Git and GitHub without any code!
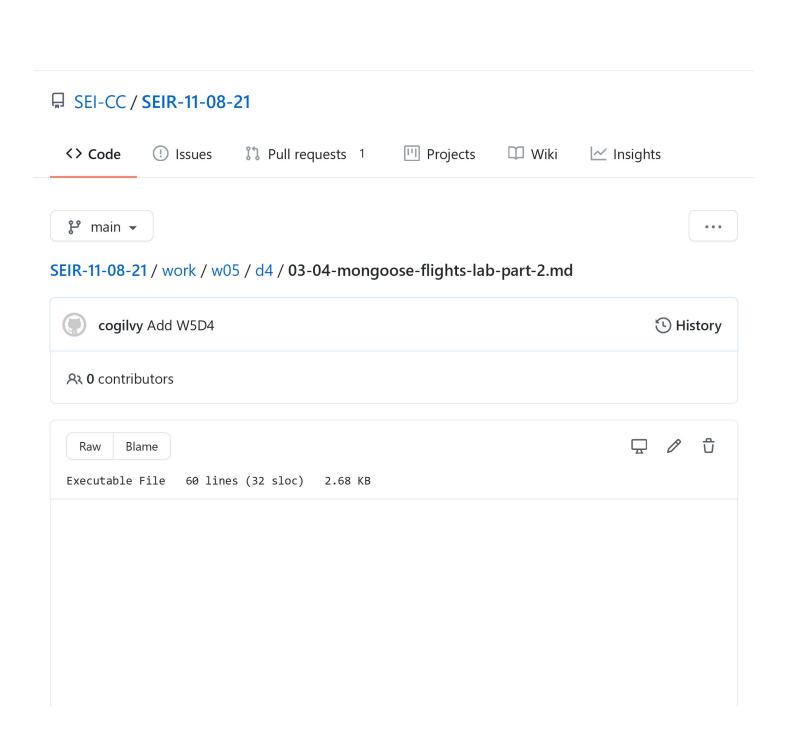
Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

`<> Code`   ⊘ Issues   ⇄ Pull requests 1   ▦ Projects   📖 Wiki   ⬚ Insights

⑂ main ▾   ⋯

SEIR-11-08-21 / work / w05 / d4 / **03-04-mongoose-flights-lab-part-2.md**

cogilvy Add W5D4   🕘 History

👥 **0 contributors**

Raw | Blame   🖥 ✎ 🗑

Executable File   60 lines (32 sloc)   2.68 KB

Mongoose "Flights" Lab - Part 2

Intro

Today in the *Mongoose - Embedding Related Data* lesson you:

- Created a schema used to embed *review* subdocuments in a *movie* document.

- Created routes and a controller for the *reviews* data resource.

- Created UI for creating and displaying the *reviews* on the **show** view of a movie.

- Wrote a `create` action that retrieved a *movie* document, pushed the *review* (`req.body`) into the document's `reviews` array, saved the *movie* doc, and redirected back to the show view for that movie.

Similar to what we did in the lesson, in this lab you'll be adding functionality to the `mongoose-flights` project you created in the *part 1* lab.

The final version of `mongoose-flights` will be a deliverable, so do each part and don't fall behind.

Goal

The goal of this lab is to add the ability to specify a list (array) of `destinations` for the flight - if the flight is non-stop, then there will be only one destination sub-doc in the array.

Styling is secondary, spend time on it only after the functionality has been implemented.

Exercises

1. Create a `destinationSchema` that will provide the structure for *destination* subdocuments with the following properties:

Property	Type	Validations	Default Value
airport	String	`enum` to include 'AUS', 'DFW', 'DEN', 'LAX' & 'SAN'	n/a
arrival	Date	n/a	n/a

2. Add the following additional property to the `Flight` Model:

Property	Type	Validations	Default Value
destinations	[destinationSchema]	n/a	n/a

3. Implement the following User Story:
   AAU, when viewing the list of flights, I want to click on a "detail" link displayed next to each flight to view all of the properties for that flight (`show` view)

4. Implement the following User Story:
   AAU, when viewing the details page (`show` view) for a flight, I want to be able to add a destination for that flight, including its `arrival` date/time & one of the established airport codes

5. Implement the following User Story:
   AAU, when viewing the details page (`show` view) for a flight, I want to see a list of that flight's `destinations` (`airport` & `arrival`)

Bonuses

1. Sort the list of `destinations` for a flight by the `arrival` date/time in ascending order.

2. Style the views.

3. When adding a destination for a flight, exclude the airports listed in the `<select>` that have already been used by other destinations and/or the flight's `airport`.

Deliverable?

The final version of `mongoose-flights` will be a deliverable, so do each part and don't fall behind.