Stephen Re – Assignment 10.1

1. Open up the Strapi Administration and create three *collection* types for your Flights API:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Steps:

1. Open Strapi in my localhost <http://localhost:1337/admin/plugins/content-type-builder/content-types/>
2. Configure each collection:
   1. Click “Create new collection type”
   2. Enter “Display Name” of collection
   3. Choose field type for first field
   4. Name first field
   5. Advanced settings > check “Required Field”
   6. Click save
3. Enter at least three rows of data for each table in your flight *collection*. Publish each *collection*. Then, make sure to change the permissions for the application plugin. Select “Select all” for each of the *collections*so that the application plugin has all the necessary permissions (count, create, delete, find, findone, and update).

FYI, these screenshots were from after everything was complete so the entries dont match. I originally included screenshots of the output from Strapi, not the browser. I left the screenshots from Strapi here as well just to show I did it.

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

A close up of text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

Steps:

1. Go to “content manager” > “collection types” > collection in question
2. Click “create new entry”
3. Fill out form
4. Click “save” then “publish
5. Go to “settings” > “Users and permissions” > “Roles” and set permissions for the three new collections to “select all”
6. Using Postman, send a GET *request*to retrieve all of the data from your Flights application. A screenshot of a computer

   Description automatically generated

Steps:

* 1. Set headers to “content-type” and “application/json” to choose the format of the request and response.
  2. Set the request for “GET” and the flights collection
  3. Send.

1. Using Postman, send a GET*request* to get the flight with id = 1. A screenshot of a computer

   Description automatically generated

Steps: Same as above, but adding the id to the end of the url retrieves just that record.

1. Using Postman, make a POST *request* to create a new flight.

A screenshot of a computer

Description automatically generated

Steps: Ran POST request on the flights collection to add new flight including the Airline, OriginalAirport, and DestinationAirport

1. Point your browser to the Flights Collection and include a screenshot of the browser to show that a new flight has been created. Include a description of your understanding of this step. A close up of text

   Description automatically generated

New flight was populated correctly including the relation fields

1. Using Postman, send a POST *request* to update a flight from your Flights application. Open Postman and enter the following URL: “http://localhost:1337/flights/”, where “” is the “id” of the flight assigned by Strapi. In the body of the request, enter text formatted as JSON with the information about the flight being changed. In this case, you are changing the airline to “American Airlines” and changing the number of seats to 60. Select PUT as the method. Select “Send.” A screenshot of a computer

   Description automatically generated

Ran PUT request to update the seat count and the airline for flight id 2.

1. Point your browser to the Flights *Collection*and include a screenshot of the browser to show that the flight has been updated.A close up of a text

   Description automatically generated

Pointed my browser to the specific record on my localhost.

1. Using Postman, send a DELETE*request* to delete a flight from your Flights application.

A screenshot of a computer

Description automatically generated

Sent DELETE request to the flights collection to delete flight id 3

1. Point your browser to the Flights *Collection* and include a screenshot of the browser to show that the flight has been deleted.

A screenshot of a computer program

Description automatically generated

Pointed my browser to the specific record on my localhost and a 404 error was returned.

1. Using Postman, obtain the code to create a Python application to make a GET *request* to your airline application. Run the code in VS Code and see the results.A screen shot of a computer

   Description automatically generated

Copied the code from Postman and ran it in vscode. The returned data contains the new American Airlines record I created for step 7