For this module’s final assignment, you will create an API for flight information. You will be using Strapi to create and publish an API, like the restaurant example in Mini-Lesson 10.7. You will also demonstrate your knowledge of*containers*and how to run code inside *containers*.

This assignment will give you the opportunity to create an API on your own. This will be useful in future modules and as part of an application for a future data engineer employee. Remember that mastering these skills is of paramount importance as you will be able to add this final assignment to your GitHub portfolio as evidence of your capabilities.

Read through the entire assignment carefully before beginning. Ensure that you understand what evidence you will need to screenshot and save for each step. At the end of this assignment, you will be asked to add all of your screenshots to a Word document to submit for grading.

To complete the assignment, perform the following steps:

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

Airline

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Type** | **Required** |
| Name | Text | Yes |

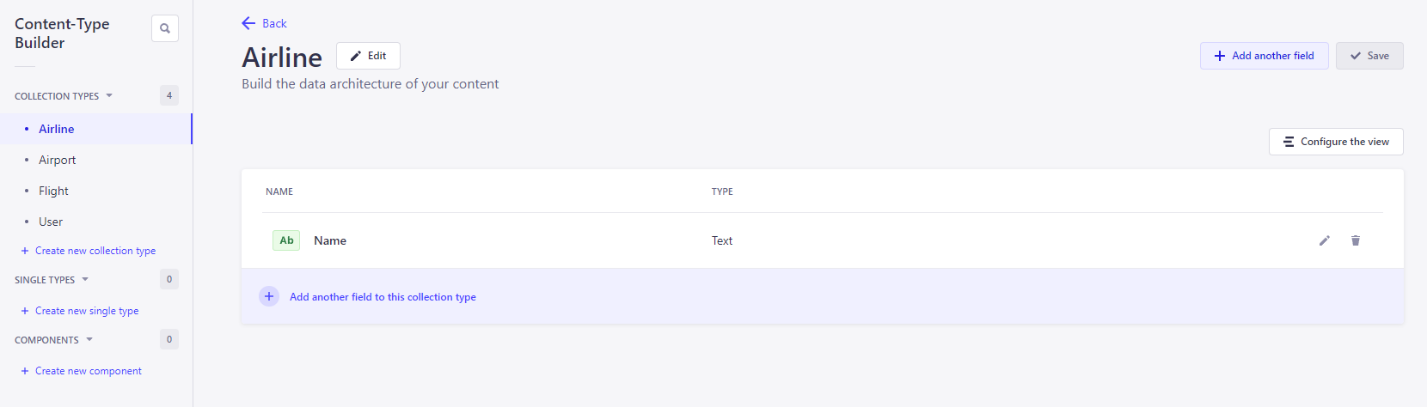
Airport

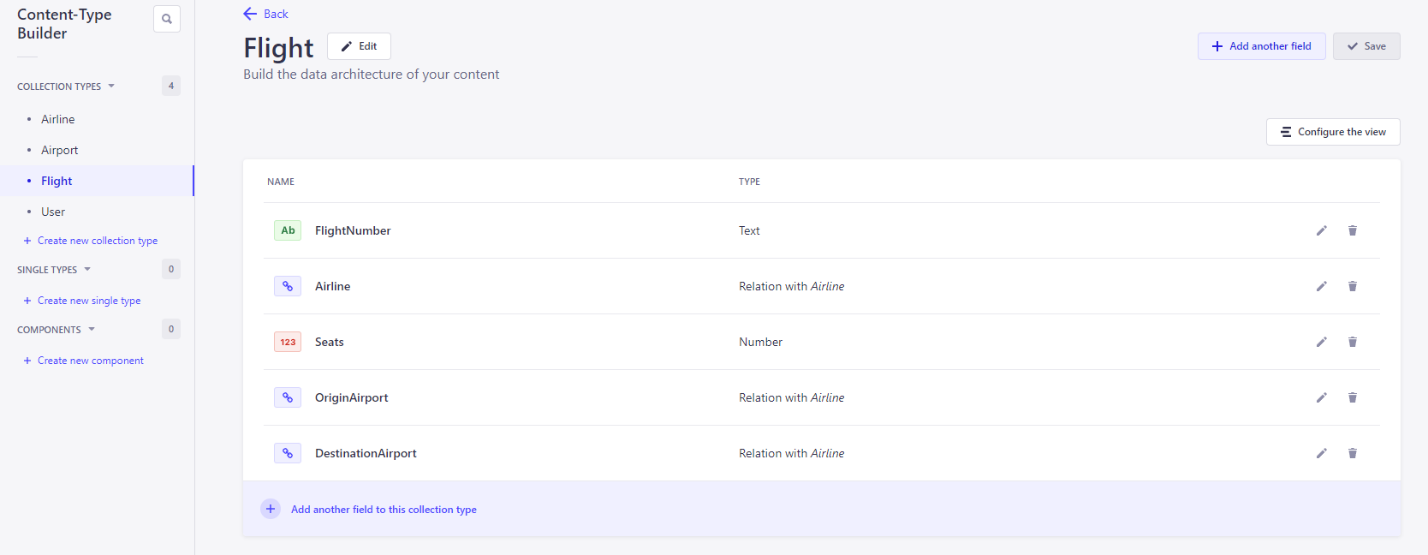
|  |  |  |
| --- | --- | --- |
| **Field Name** | **Type** | **Required** |
| AirportCode | Text | Yes |
| AirportName | Text | Yes |
| Country | Text | Yes |
| State | Text | Yes |
| City | Text | Yes |

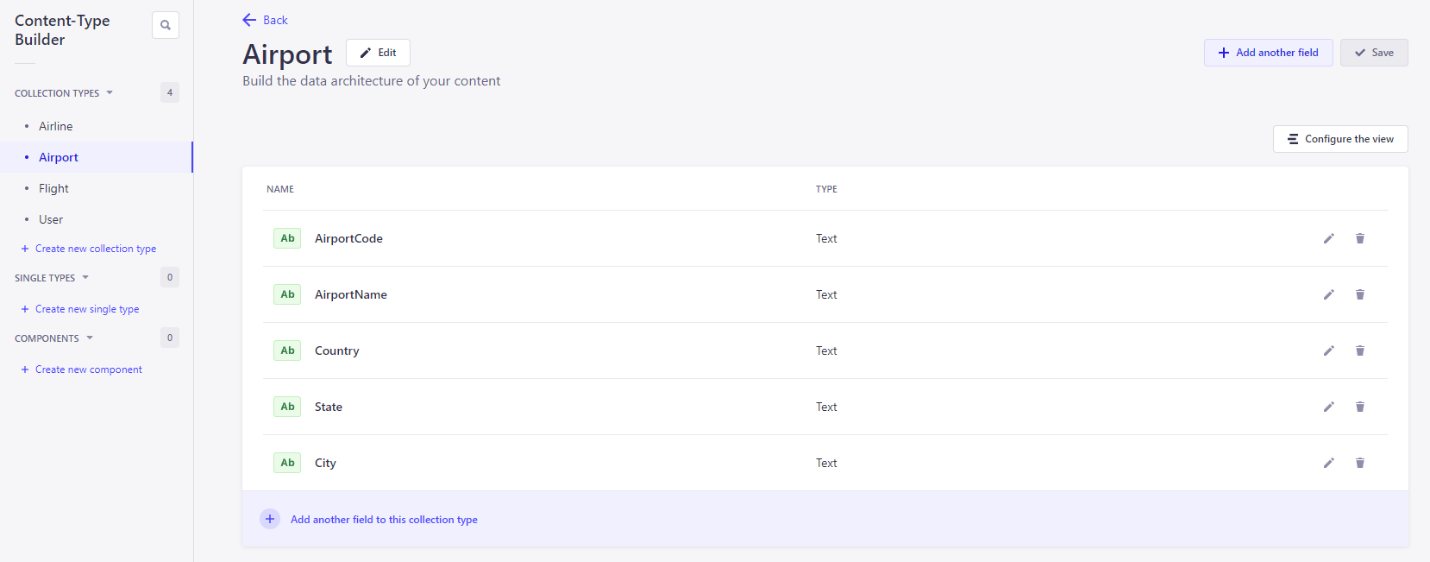
Flight

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Type** | **Required** |
| FlightNumber | Text | Yes |
| Airline | Relation to *Collection* Type Airline. Flight has one airline. | Yes |
| Seats | Number. Number format: Integer | Yes |
| OriginAirport | Relation to *Collection* Type Airport. Flight origin has one airport. | Yes |
| DestinationAirport | Relation to *Collection* Type Airport. Flight destination has one airport. | Yes |

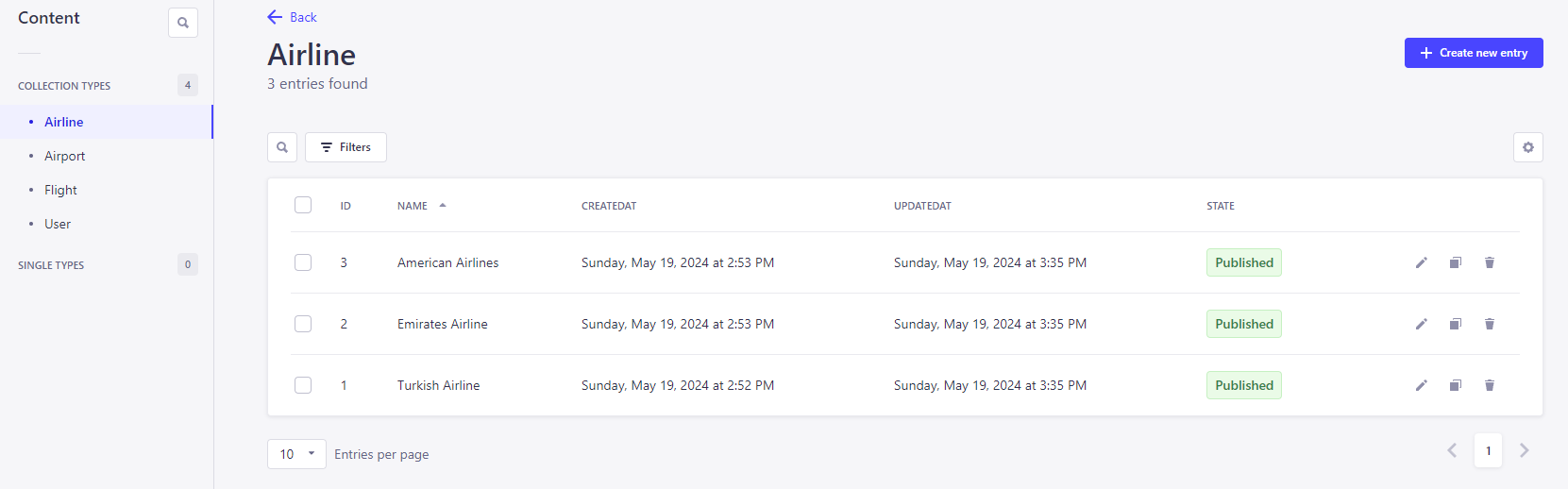
Take a screenshot of each of the three*collections* after you create them in Strapi. Add these screenshots to a Word document and label them. Save your Word document on your computer after each step. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

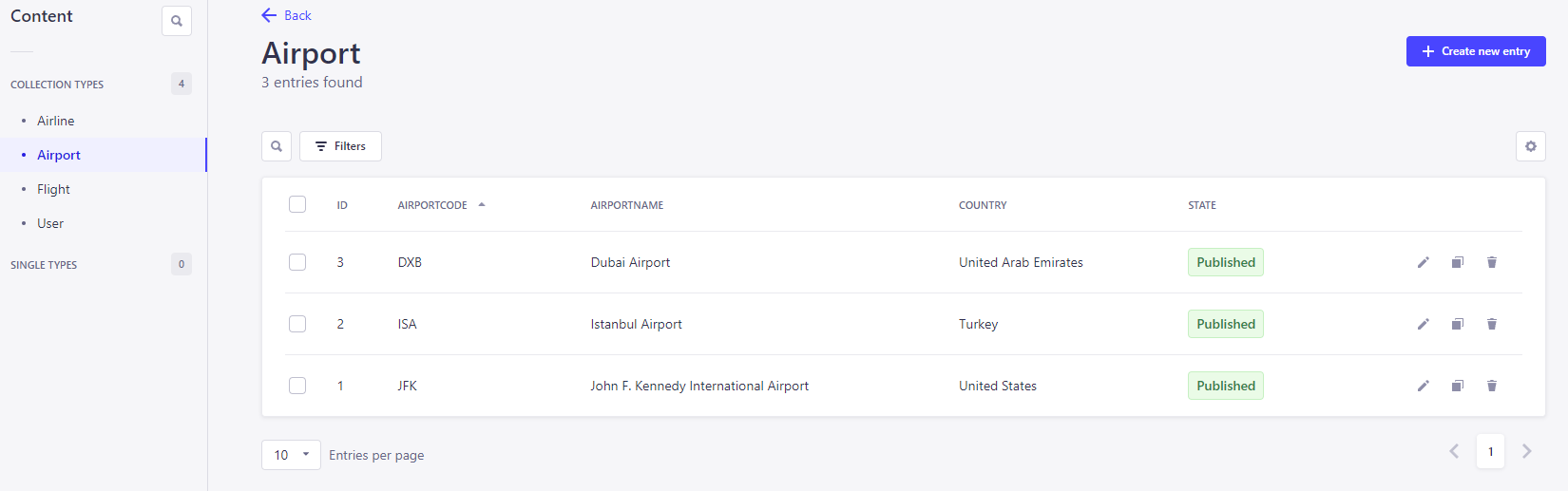


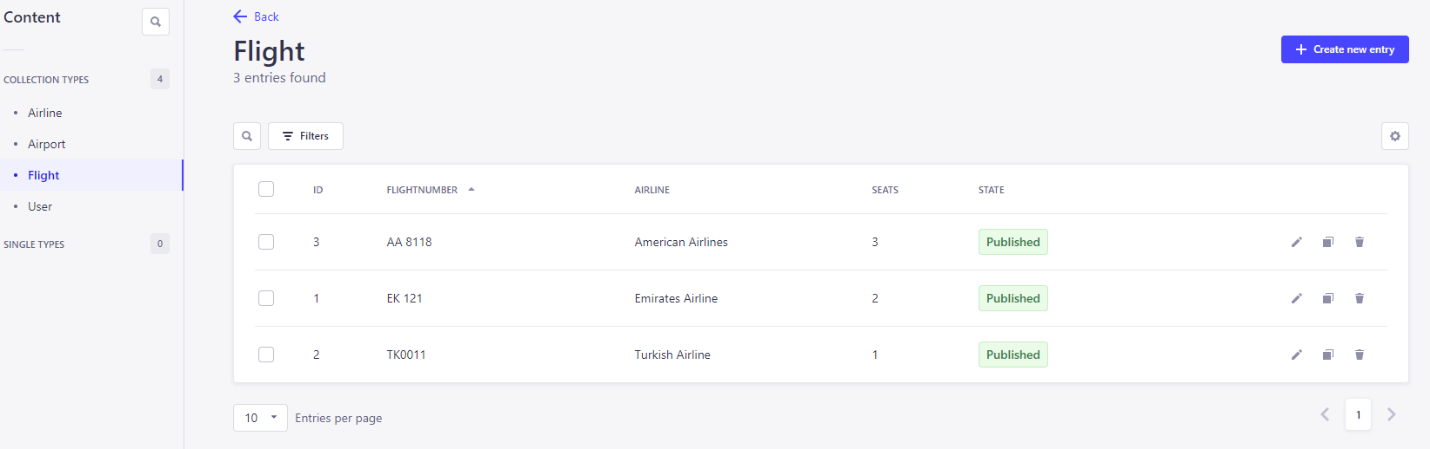


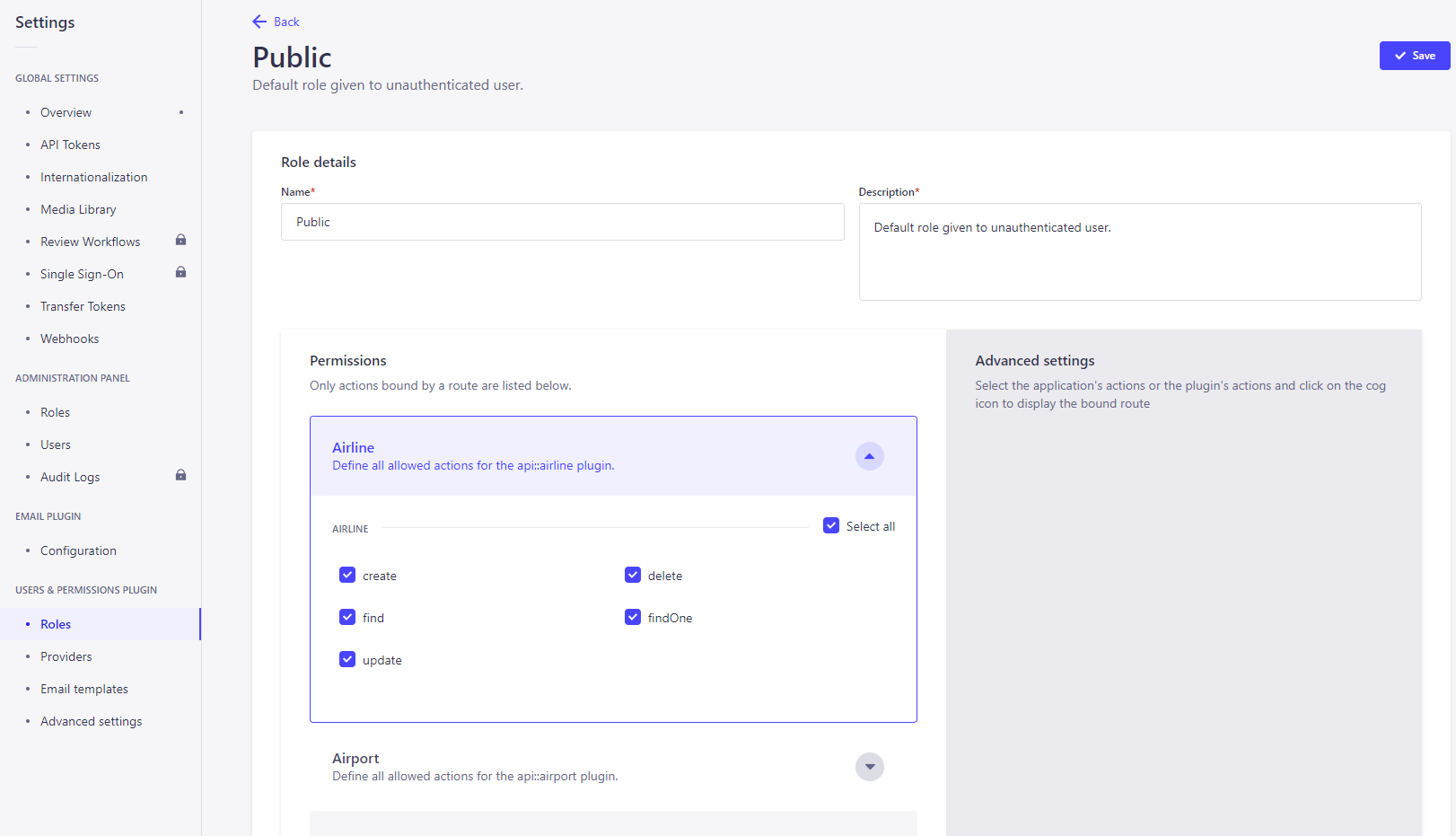


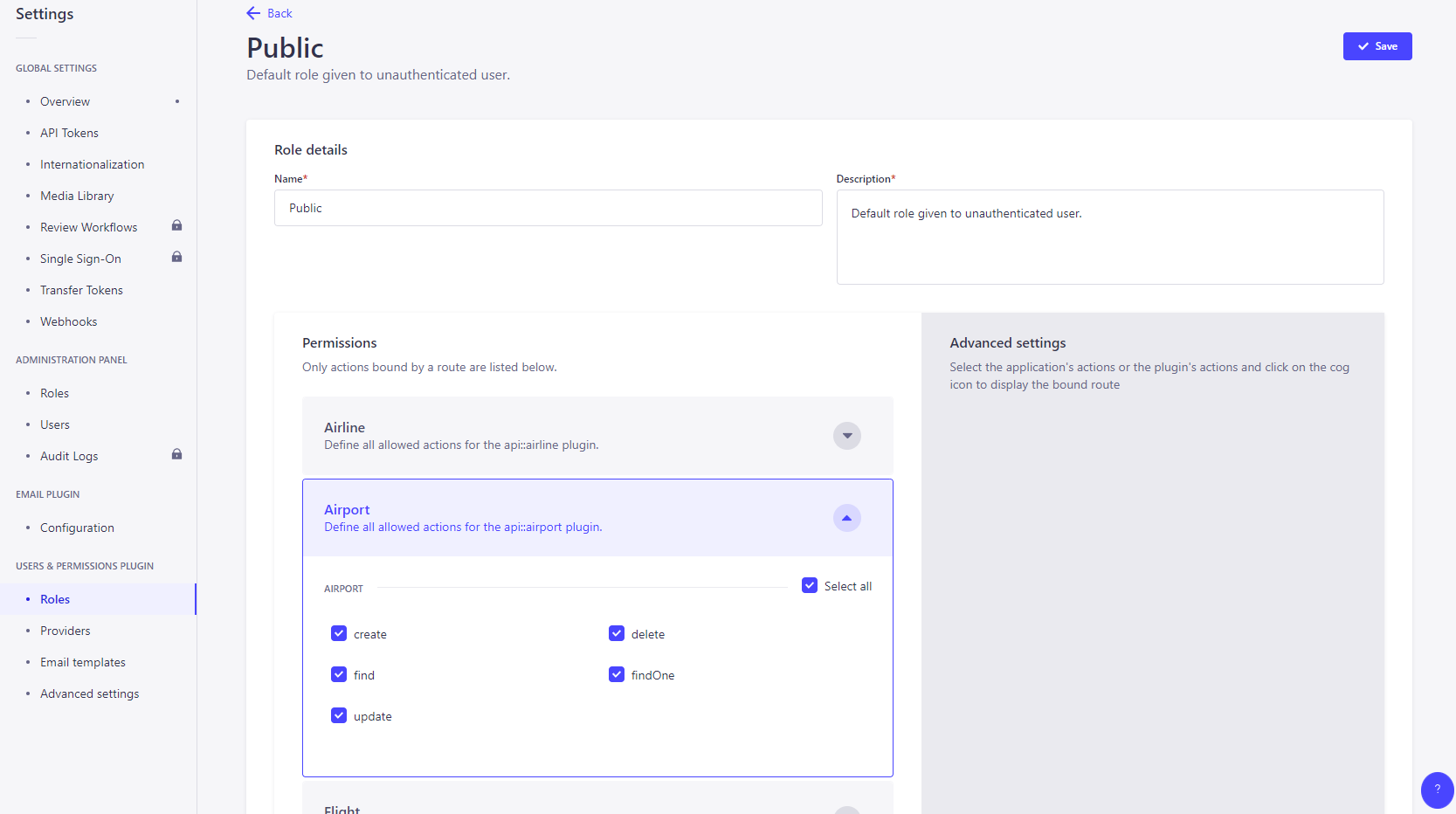
1. Enter at least three rows of data for each table in your flight *collection*. Publish each *collection*. Then, make sure to change the permissions for the application plugin. Select “Select all” for each of the *collections*so that the application plugin has all the necessary permissions (count, create, delete, find, findone, and update).

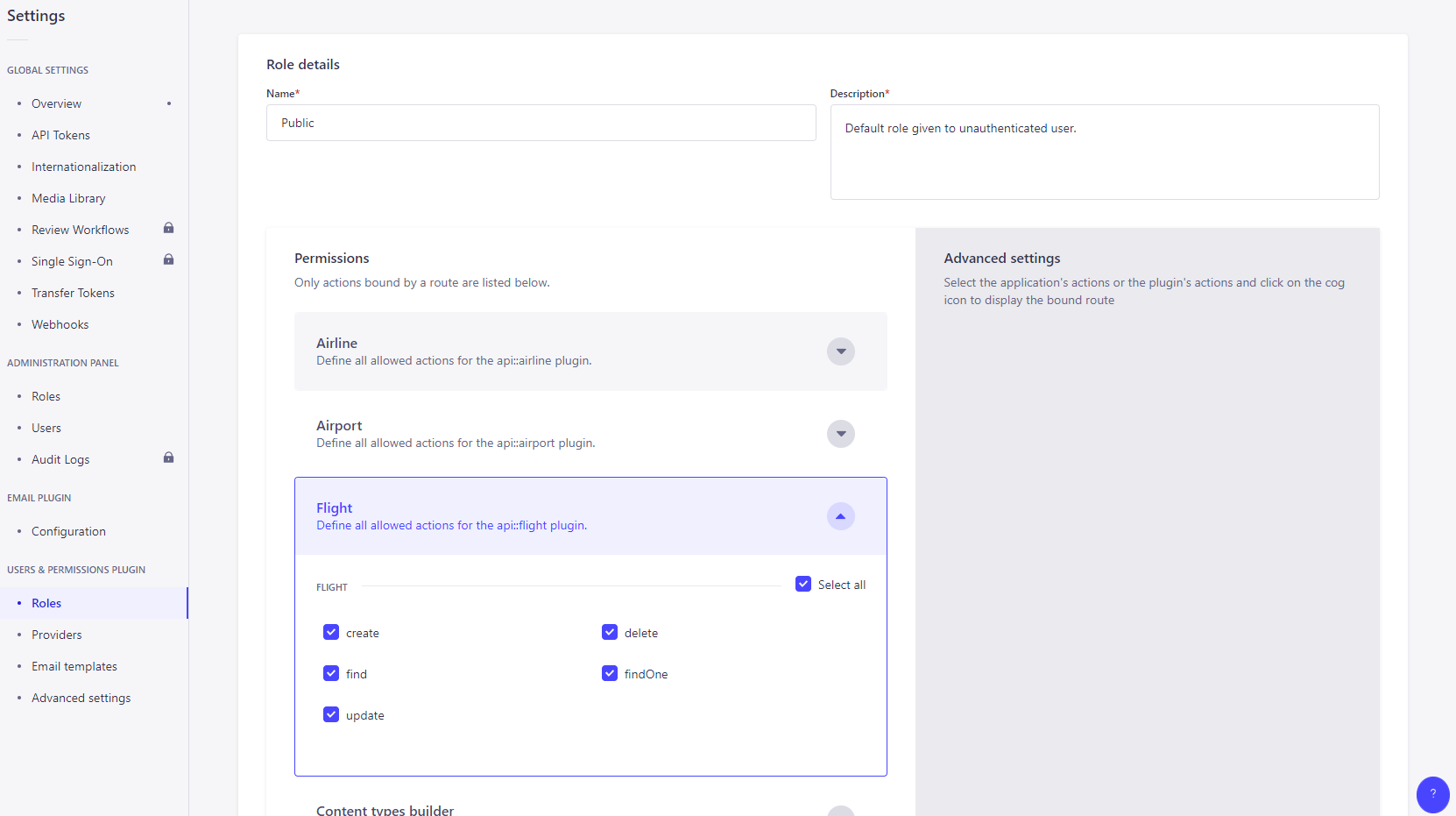












Now, navigate to your browser and enter the URL for each of the collections so that they can be displayed as follows:

Flights *Collection*, Airports *Collection*, Airlines *Collection*.

Show evidence that you have created data for the Flights API by pointing your browser to each of the collection types and taking a screenshot of the output. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

1. Using Postman, send a GET *request*to retrieve all of the data from your Flights application.

Show evidence that you have submitted the GET *request* by taking a screenshot of the *request* and response in Postman. Add these screenshots to your Word document and label them. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

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

Show evidence that you have submitted the GET*request*by taking a screenshot of the *request* and response in Postman. Then, add it to your Word document and label it. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

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

Show evidence that you have submitted the POST *request* by taking a screenshot of the *request* and response in Postman. Then, add it to your Word document and label it. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

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

Show evidence that you have submitted the POST *request* by taking a screenshot of the *request* and response in Postman. Add the screenshot to your Word document and label it. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

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.”

Show evidence that you have submitted the POST *request* by taking a screenshot of the *request* and response in Postman. Add the screenshot to your Word document and label it. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

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

Show evidence that you have submitted the POST *request* by taking a screenshot of the *request* and response in Postman. Add the screenshot to your Word document and label it. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

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

Show evidence that you have submitted the DELETE *request*by taking a screenshot of the *request*and response in Postman. Add the screenshot to your Word document and label it. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

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

Show evidence that you have submitted the POST *request* by taking a screenshot of the *request* and response in Postman. Add the screenshot to your Word document and label it. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

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.

Show evidence that you have completed these steps by creating screenshots of Strapi and Postman as well as the output. Add the screenshots to your Word document and label them. Ensure that you describe each of the steps in your document to demonstrate your understanding of the steps that you have performed so far.

**Submission Instructions:**

Your submission for this assignment should be a Word document that includes the following screenshots, each labeled for the step that the screenshot represents. For each of the steps below, ensure that you also include a description of the steps in your document to demonstrate your understanding of the steps that you have performed so far:

1. Show evidence that the three*collections* have been created in Strapi by producing screenshots for each of the *collection* types after they are created. Include a description of your understanding of this step.
2. Show evidence that you have created data for the Flights API by pointing your browser to each of the *collection*types and taking a screenshot of the output. Include a description of your understanding of this step.
3. Show evidence that you have submitted the GET *request* by taking a screenshot of the *request* and response in Postman. Include a description of your understanding of this step.
4. Show evidence that you have submitted the GET*request*by taking a screenshot of the *request* and response in Postman. Include a description of your understanding of this step.
5. Show evidence that you have submitted the POST *request* by taking a screenshot of the *request* and response in Postman. Include a description of your understanding of this step.
6. 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.
7. Show evidence that you have submitted the POST *request* by taking a screenshot of the *request* and response in Postman. Include a description of your understanding of this step.
8. Point your browser to the Flights *Collection* and include a screenshot of the browser to show that flight has been updated. Include a description of your understanding of this step.
9. Show evidence that you have submitted the DELETE *request*by taking a screenshot of the *request*and response in Postman. Include a description of your understanding of this step.
10. Point your browser to the Flights *Collection*and include a screenshot of the browser to show that the flight has been deleted. Include a description of your understanding of this step.
11. Run the code to create a Python application to make a GET *request* to your airline application. Show evidence that you have completed these steps by creating screenshots of Strapi and Postman as well as the output. Include a description of your understanding of this step.

* Estimated time: 2 hrs
* *This is a required assignment and counts toward course completion.*