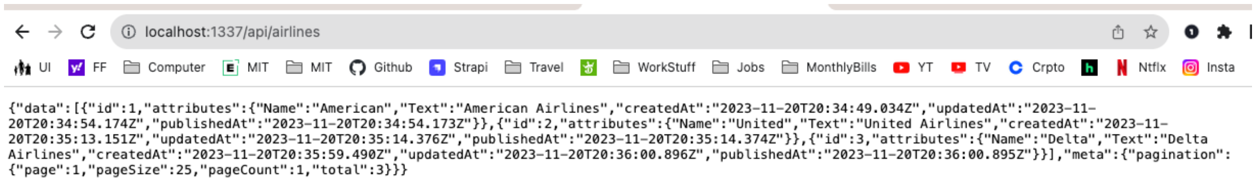
Module 10 Final Assignment

Stephen Barnes

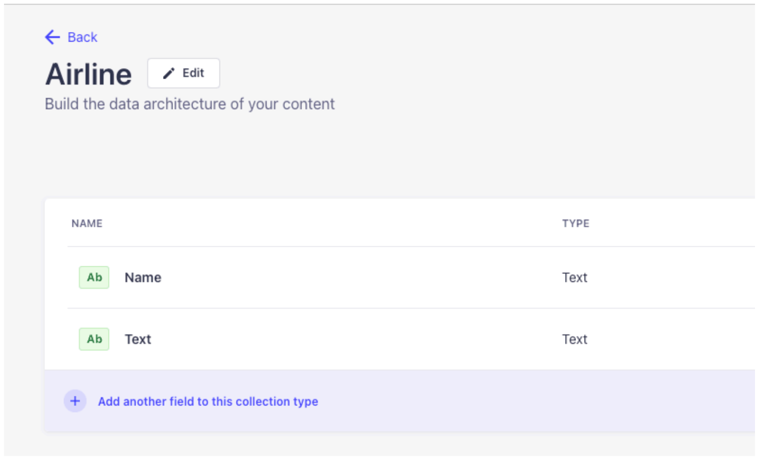
Please see below the three *collection* types for the Flights API. These were created by following Professor William's instructions and by watching YouTube tutorials at https://www.youtube.com/@Strapi. Basically, the steps were to create the collections, add their respective fields, define those fields (including relationships to other collections, if applicable) and establish permissions and authorizations.

Airlines: Here, you can see in my browser the API for Airlines…


localhost:1337/api/airlines
UI FF Computer MIT MIT Github Strapi Travel WorkStuff Jobs MonthlyBills YT TV Crpto Ntflx Insta

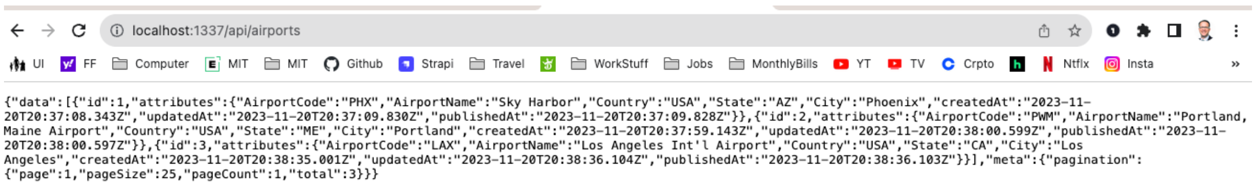
```
{"data":[{"id":1,"attributes":{"Name":"American","Text":"American Airlines","createdAt":"2023-11-20T20:34:49.034Z","updatedAt":"2023-11-
20T20:34:54.174Z","publishedAt":"2023-11-20T20:34:54.173Z"}},{"id":2,"attributes":{"Name":"United","Text":"United Airlines","createdAt":"2023-11-
20T20:35:13.151Z","updatedAt":"2023-11-20T20:35:14.376Z","publishedAt":"2023-11-20T20:35:14.374Z"}},{"id":3,"attributes":{"Name":"Delta","Text":"Delta
Airlines","createdAt":"2023-11-20T20:35:59.490Z","updatedAt":"2023-11-20T20:36:00.896Z","publishedAt":"2023-11-20T20:36:00.895Z"}}],"meta":{"pagination":
{"page":1,"pageSize":25,"pageCount":1,"total":3}}}
```

and in Strapi's UI.


← Back
Airline
Edit
Build the data architecture of your content
Add another field to this collection type

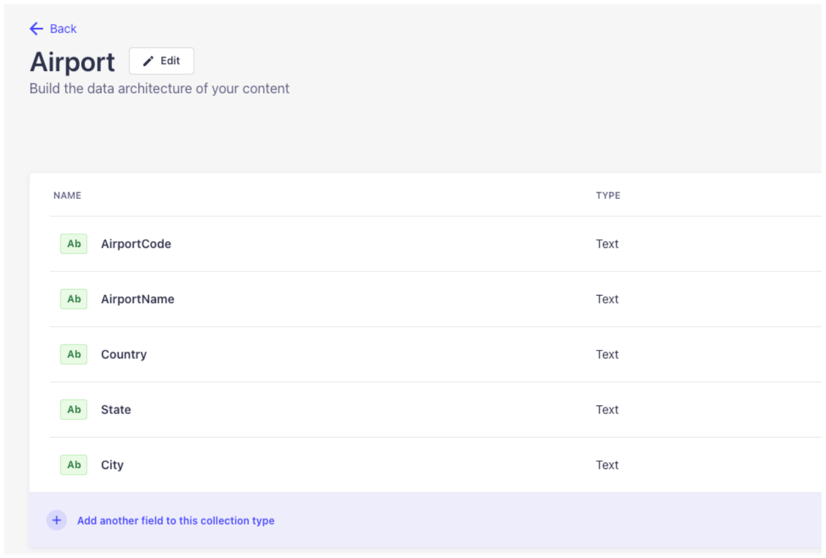
NAME	TYPE
Ab Name	Text
Ab Text	Text

Likewise, here is the **Airport** info:


localhost:1337/api/airports
UI FF Computer MIT MIT Github Strapi Travel WorkStuff Jobs MonthlyBills YT TV Crpto Ntflx Insta

```
{"data":[{"id":1,"attributes":{"AirportCode":"PHX","AirportName":"Sky Harbor","Country":"USA","State":"AZ","City":"Phoenix","createdAt":"2023-11-
20T20:37:08.343Z","updatedAt":"2023-11-20T20:37:09.830Z","publishedAt":"2023-11-20T20:37:09.828Z"}},{"id":2,"attributes":{"AirportCode":"PWM","AirportName":"Portland,
Maine Airport","Country":"USA","State":"ME","City":"Portland","createdAt":"2023-11-20T20:37:59.143Z","updatedAt":"2023-11-20T20:38:00.599Z","publishedAt":"2023-11-
20T20:38:00.597Z"}},{"id":3,"attributes":{"AirportCode":"LAX","AirportName":"Los Angeles Int'l Airport","Country":"USA","State":"CA","City":"Los
Angeles","createdAt":"2023-11-20T20:38:35.001Z","updatedAt":"2023-11-20T20:38:36.104Z","publishedAt":"2023-11-20T20:38:36.103Z"}}],"meta":{"pagination":
{"page":1,"pageSize":25,"pageCount":1,"total":3}}}
```

As well as in Strapi:



Back

Airport

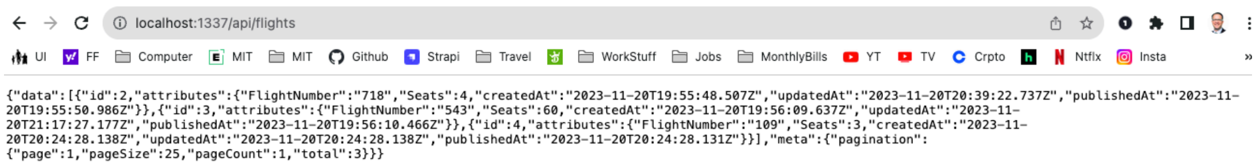
Edit

Build the data architecture of your content

NAME	TYPE
Ab AirportCode	Text
Ab AirportName	Text
Ab Country	Text
Ab State	Text
Ab City	Text

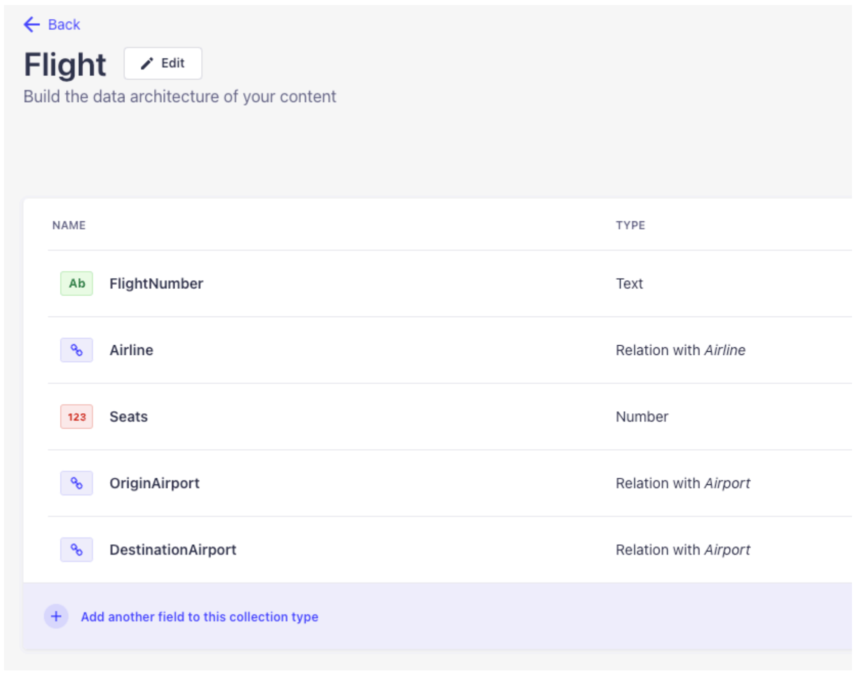
+ Add another field to this collection type

And, finally, here is the **<u>Flight</u>** info:



```
{"data":[{"id":2,"attributes":{"FlightNumber":"718","Seats":4,"createdAt":"2023-11-20T19:55:48.507Z","updatedAt":"2023-11-20T20:39:22.737Z","publishedAt":"2023-11-
20T19:55:50.986Z"}},{"id":3,"attributes":{"FlightNumber":"543","Seats":60,"createdAt":"2023-11-20T19:56:09.637Z","updatedAt":"2023-11-
20T21:17:27.177Z","publishedAt":"2023-11-20T19:56:10.466Z"}},{"id":4,"attributes":{"FlightNumber":"109","Seats":3,"createdAt":"2023-11-
20T20:24:28.138Z","updatedAt":"2023-11-20T20:24:28.138Z","publishedAt":"2023-11-20T20:24:28.131Z"}}],"meta":{"pagination":
{"page":1,"pageSize":25,"pageCount":1,"total":3}}}
```

Strapi:



Back

Flight

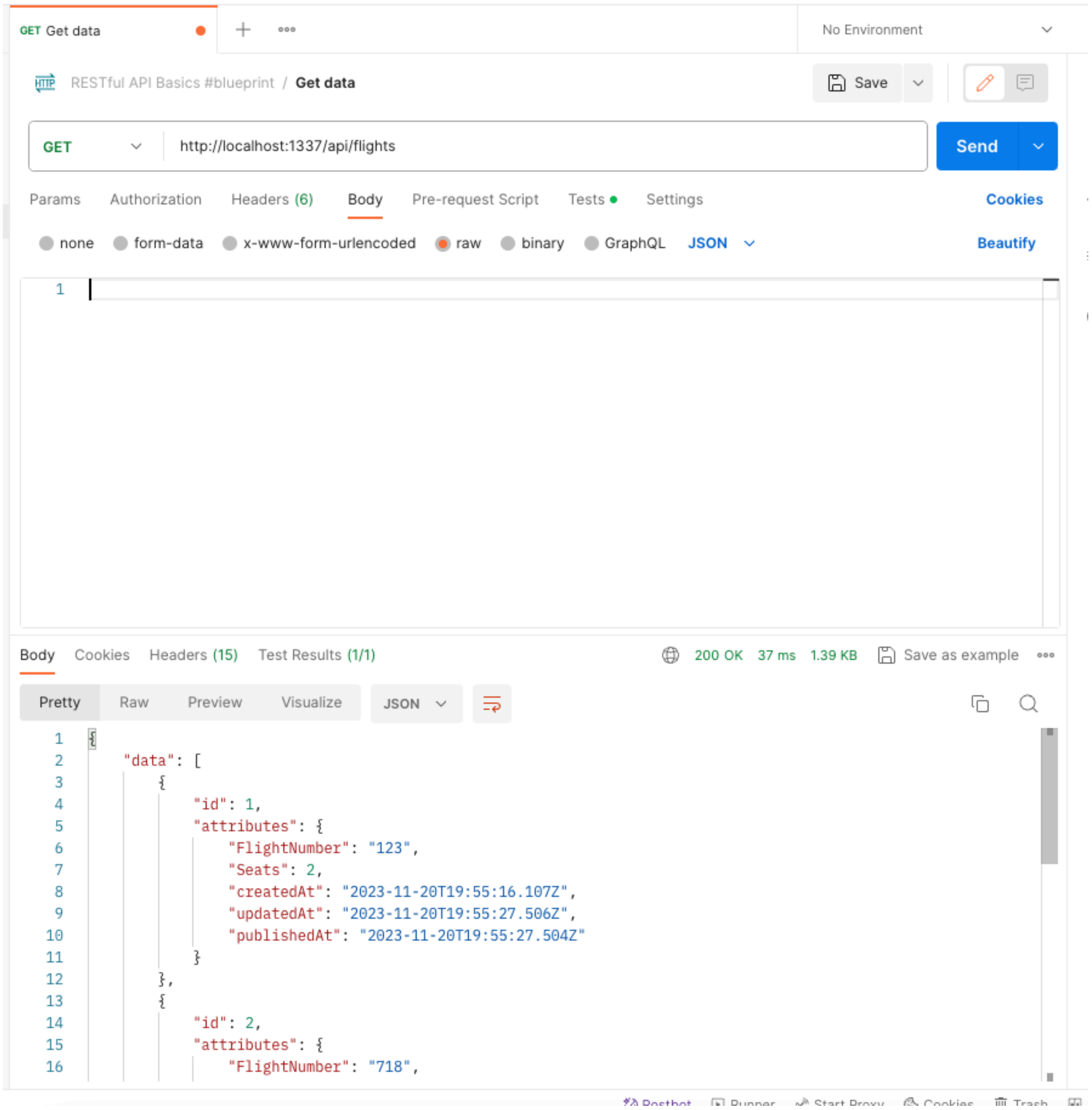
Edit

Build the data architecture of your content

NAME	TYPE
Ab FlightNumber	Text
Airline	Relation with *Airline*
123 Seats	Number
OriginAirport	Relation with *Airport*
DestinationAirport	Relation with *Airport*

+ Add another field to this collection type

As evidenced above, I included data for each of the collections. Then, I did a GET request in Postman to retrieve the 'Flights' information. As you can see below, the request worked ("200 OK") and fetched info for all flights.

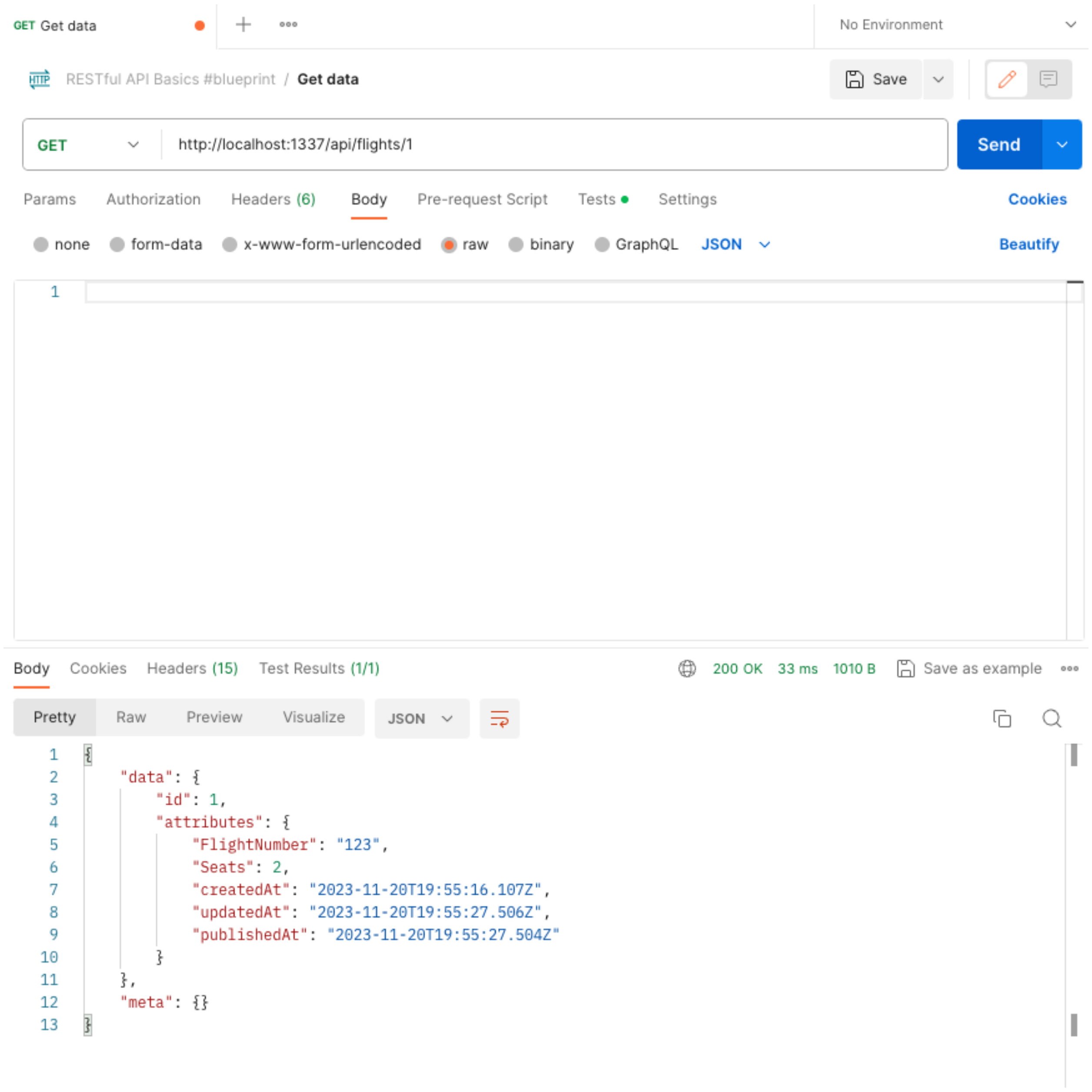


GET http://localhost:1337/api/flights

200 OK 37 ms 1.39 KB

```
{
    "data": [
        {
            "id": 1,
            "attributes": {
                "FlightNumber": "123",
                "Seats": 2,
                "createdAt": "2023-11-20T19:55:16.107Z",
                "updatedAt": "2023-11-20T19:55:27.506Z",
                "publishedAt": "2023-11-20T19:55:27.504Z"
            }
        },
        {
            "id": 2,
            "attributes": {
                "FlightNumber": "718",
```

Then, I did another GET request for just the first 'Flights' entry:

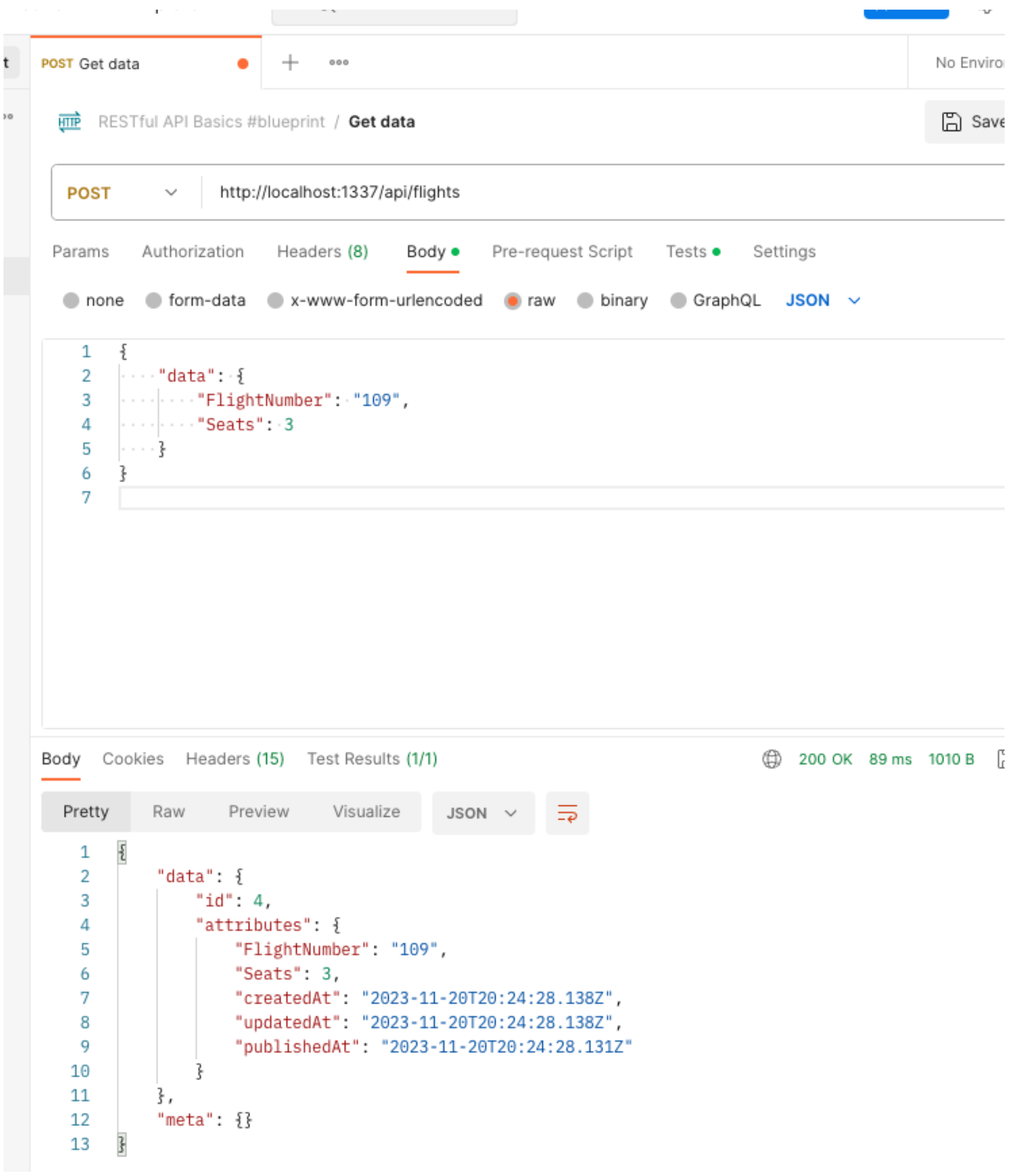


```
http://localhost:1337/api/flights/1
```

200 OK 33 ms 1010 B

```
{
    "data": {
        "id": 1,
        "attributes": {
            "FlightNumber": "123",
            "Seats": 2,
            "createdAt": "2023-11-20T19:55:16.107Z",
            "updatedAt": "2023-11-20T19:55:27.506Z",
            "publishedAt": "2023-11-20T19:55:27.504Z"
        }
    },
    "meta": {}
}
```

And then used Postman to create a new flight with a POST API.


POST Get data

No Enviro

RESTful API Basics #blueprint / Get data

Save

POST http://localhost:1337/api/flights

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

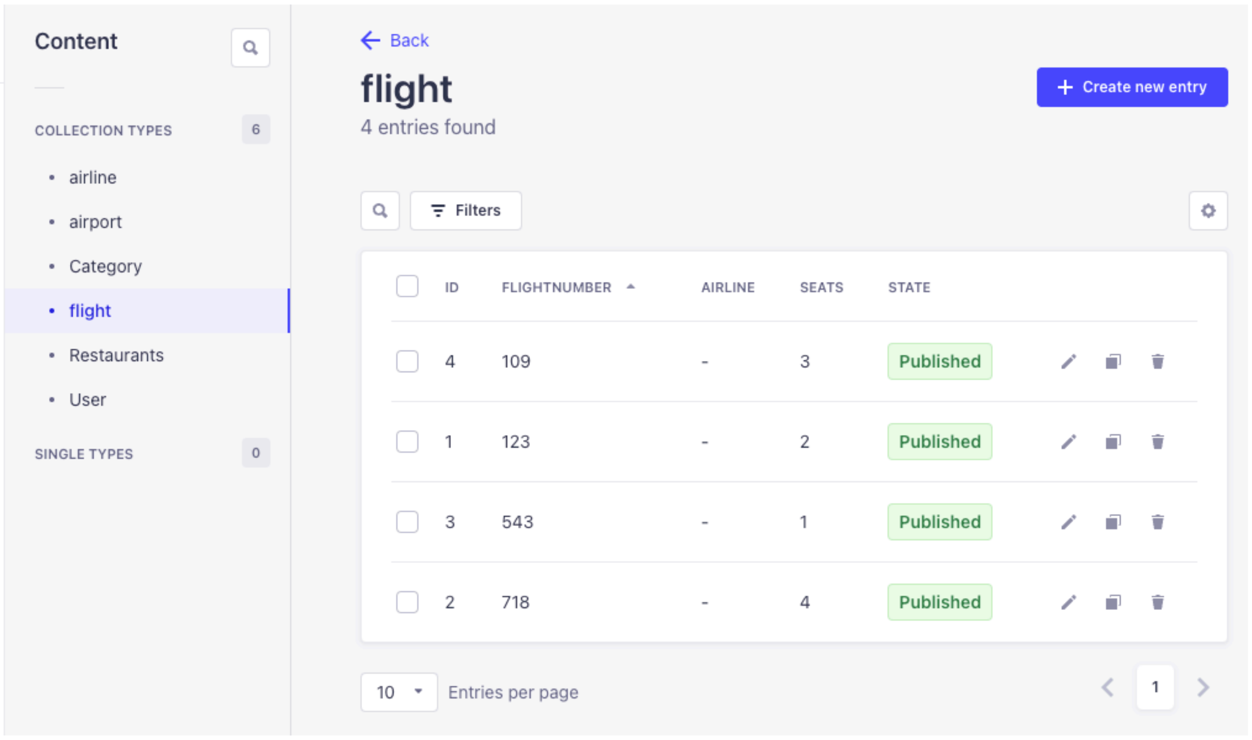
```
{
    "data": {
        "FlightNumber": "109",
        "Seats": 3
    }
}
```

Body Cookies Headers (15) Test Results (1/1) 200 OK 89 ms 1010 B

Pretty Raw Preview Visualize JSON

```
{
    "data": {
        "id": 4,
        "attributes": {
            "FlightNumber": "109",
            "Seats": 3,
            "createdAt": "2023-11-20T20:24:28.138Z",
            "updatedAt": "2023-11-20T20:24:28.138Z",
            "publishedAt": "2023-11-20T20:24:28.131Z"
        }
    },
    "meta": {}
}
```

Which then appeared in my CMS, Strapi:



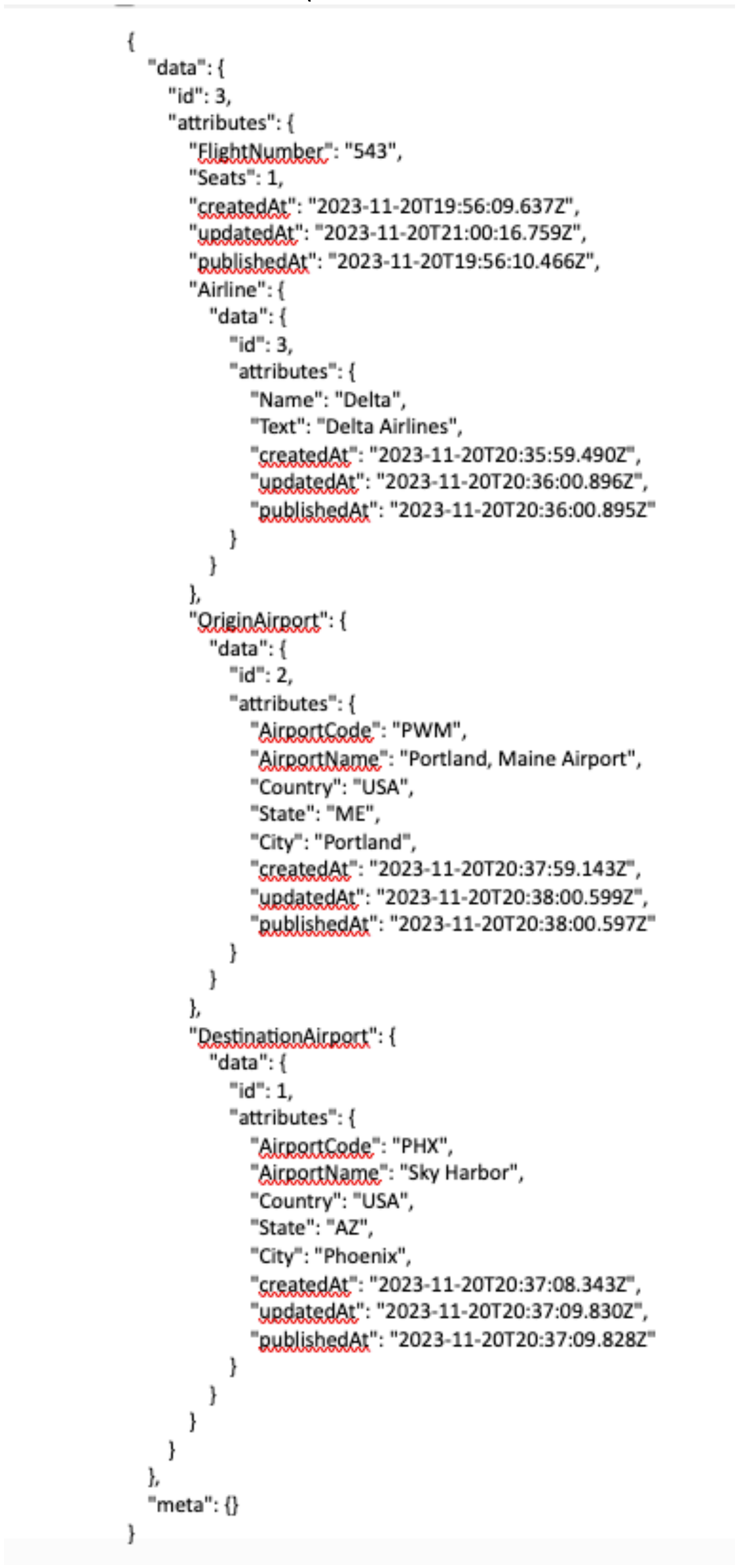
flight

4 entries found

ID	FLIGHTNUMBER	AIRLINE	SEATS	STATE
4	109	-	3	Published
1	123	-	2	Published
3	543	-	1	Published
2	718	-	4	Published

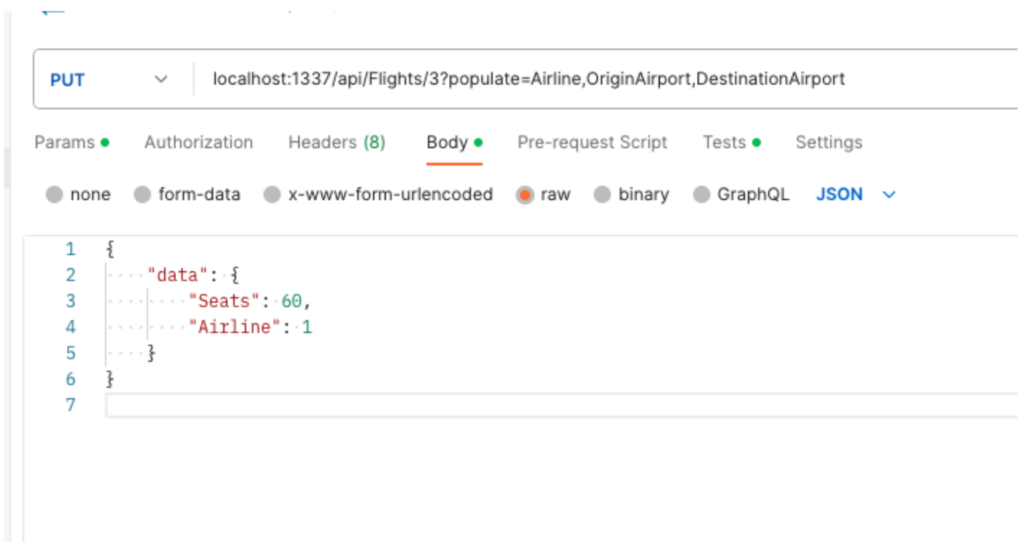
After using Postman to create a new flight, I used a PUT API in Postman to update flight 3. The screenshot below is the 'before' code that was returned using a GET API for flight 3. Further down, you'll see the code, and the requested changes that was returned after I did the PUT API.

BEFORE THE CHANGES (this was done with a GET API in Postman):



```
{
  "data": {
    "id": 3,
    "attributes": {
      "FlightNumber": "543",
      "Seats": 1,
      "createdAt": "2023-11-20T19:56:09.637Z",
      "updatedAt": "2023-11-20T21:00:16.759Z",
      "publishedAt": "2023-11-20T19:56:10.466Z",
      "Airline": {
        "data": {
          "id": 3,
          "attributes": {
            "Name": "Delta",
            "Text": "Delta Airlines",
            "createdAt": "2023-11-20T20:35:59.490Z",
            "updatedAt": "2023-11-20T20:36:00.896Z",
            "publishedAt": "2023-11-20T20:36:00.895Z"
          }
        }
      },
      "OriginAirport": {
        "data": {
          "id": 2,
          "attributes": {
            "AirportCode": "PWM",
            "AirportName": "Portland, Maine Airport",
            "Country": "USA",
            "State": "ME",
            "City": "Portland",
            "createdAt": "2023-11-20T20:37:59.143Z",
            "updatedAt": "2023-11-20T20:38:00.599Z",
            "publishedAt": "2023-11-20T20:38:00.597Z"
          }
        }
      },
      "DestinationAirport": {
        "data": {
          "id": 1,
          "attributes": {
            "AirportCode": "PHX",
            "AirportName": "Sky Harbor",
            "Country": "USA",
            "State": "AZ",
            "City": "Phoenix",
            "createdAt": "2023-11-20T20:37:08.343Z",
            "updatedAt": "2023-11-20T20:37:09.830Z",
            "publishedAt": "2023-11-20T20:37:09.828Z"
          }
        }
      }
    }
  },
  "meta": {}
}
```

AFTER THE CHANGES (this was done with a PUT API in Postman):

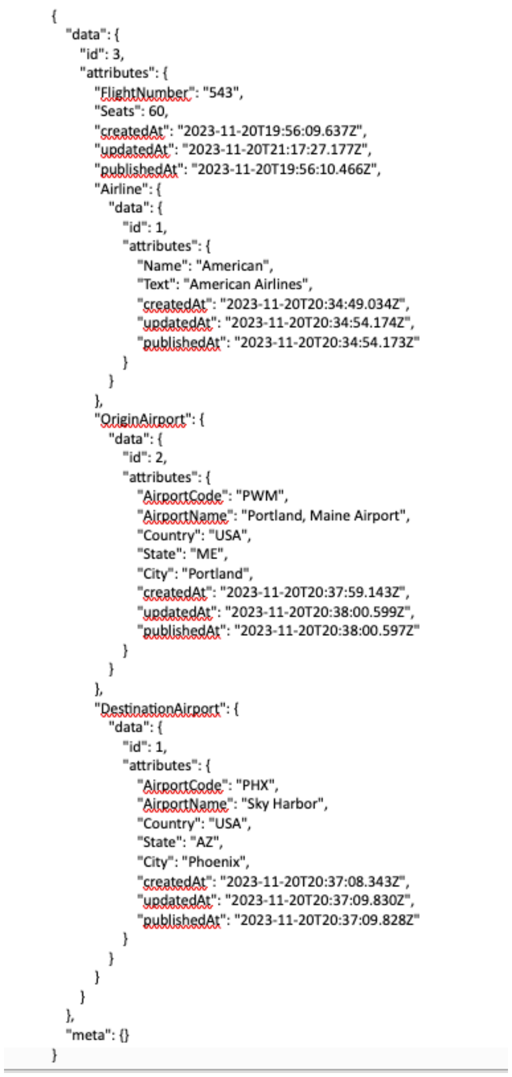


PUT localhost:1337/api/Flights/3?populate=Airline,OriginAirport,DestinationAirport

Params ● Authorization Headers (8) Body ● Pre-request Script Tests ● Settings

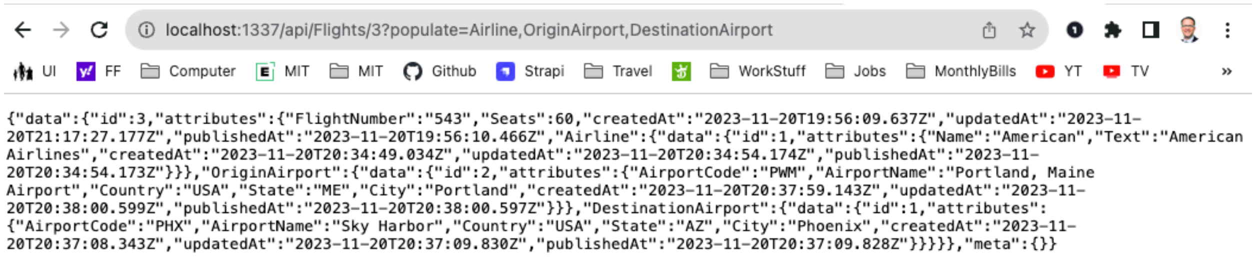
none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

```
{
    "data": {
        "Seats": 60,
        "Airline": 1
    }
}
```



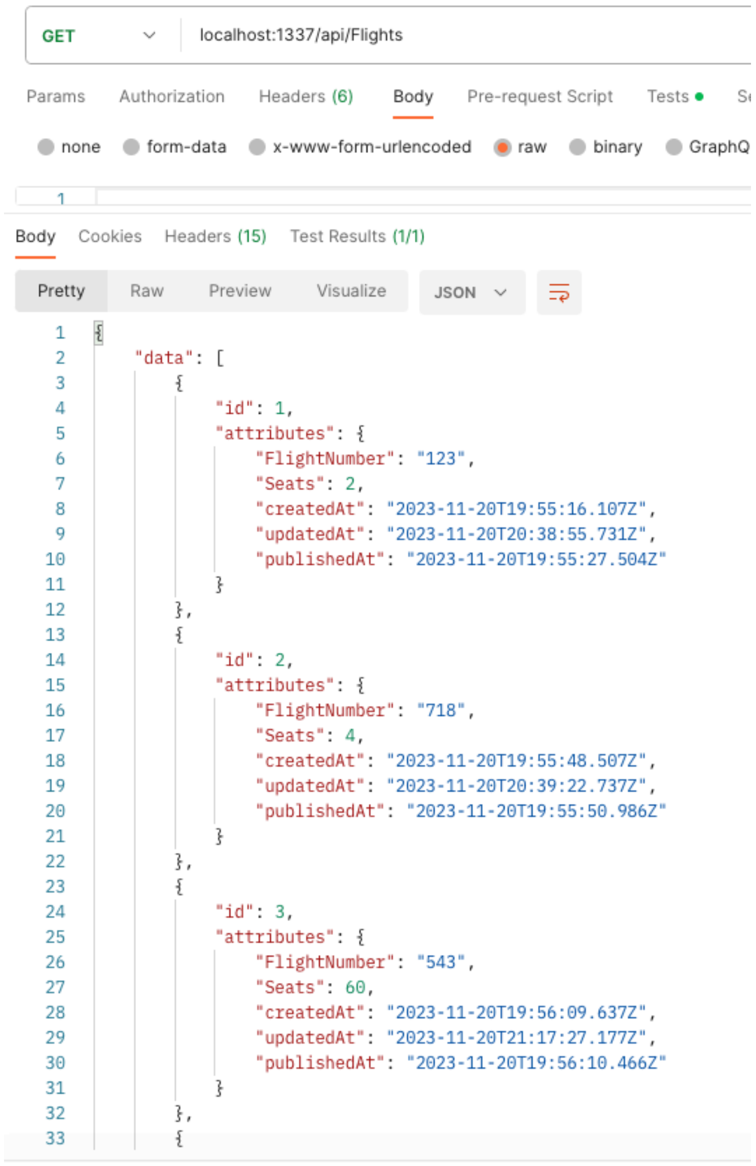
```
{
  "data": {
    "id": 3,
    "attributes": {
      "FlightNumber": "543",
      "Seats": 60,
      "createdAt": "2023-11-20T19:56:09.637Z",
      "updatedAt": "2023-11-20T21:17:27.177Z",
      "publishedAt": "2023-11-20T19:56:10.466Z",
      "Airline": {
        "data": {
          "id": 1,
          "attributes": {
            "Name": "American",
            "Text": "American Airlines",
            "createdAt": "2023-11-20T20:34:49.034Z",
            "updatedAt": "2023-11-20T20:34:54.174Z",
            "publishedAt": "2023-11-20T20:34:54.173Z"
          }
        }
      },
      "OriginAirport": {
        "data": {
          "id": 2,
          "attributes": {
            "AirportCode": "PWM",
            "AirportName": "Portland, Maine Airport",
            "Country": "USA",
            "State": "ME",
            "City": "Portland",
            "createdAt": "2023-11-20T20:37:59.143Z",
            "updatedAt": "2023-11-20T20:38:00.599Z",
            "publishedAt": "2023-11-20T20:38:00.597Z"
          }
        }
      },
      "DestinationAirport": {
        "data": {
          "id": 1,
          "attributes": {
            "AirportCode": "PHX",
            "AirportName": "Sky Harbor",
            "Country": "USA",
            "State": "AZ",
            "City": "Phoenix",
            "createdAt": "2023-11-20T20:37:08.343Z",
            "updatedAt": "2023-11-20T20:37:09.830Z",
            "publishedAt": "2023-11-20T20:37:09.828Z"
          }
        }
      }
    }
  },
  "meta": {}
}
```

And here is the update in the url as well:


localhost:1337/api/Flights/3?populate=Airline,OriginAirport,DestinationAirport

```
{"data":{"id":3,"attributes":{"FlightNumber":"543","Seats":60,"createdAt":"2023-11-20T19:56:09.637Z","updatedAt":"2023-11-
20T21:17:27.177Z","publishedAt":"2023-11-20T19:56:10.466Z","Airline":{"data":{"id":1,"attributes":{"Name":"American","Text":"American
Airlines","createdAt":"2023-11-20T20:34:49.034Z","updatedAt":"2023-11-20T20:34:54.174Z","publishedAt":"2023-11-
20T20:34:54.173Z"}}},"OriginAirport":{"data":{"id":2,"attributes":{"AirportCode":"PWM","AirportName":"Portland, Maine
Airport","Country":"USA","State":"ME","City":"Portland","createdAt":"2023-11-20T20:37:59.143Z","updatedAt":"2023-11-
20T20:38:00.599Z","publishedAt":"2023-11-20T20:38:00.597Z"}}},"DestinationAirport":{"data":{"id":1,"attributes":
{"AirportCode":"PHX","AirportName":"Sky Harbor","Country":"USA","State":"AZ","City":"Phoenix","createdAt":"2023-11-
20T20:37:08.343Z","updatedAt":"2023-11-20T20:37:09.830Z","publishedAt":"2023-11-20T20:37:09.828Z"}}}}},"meta":{}}
```

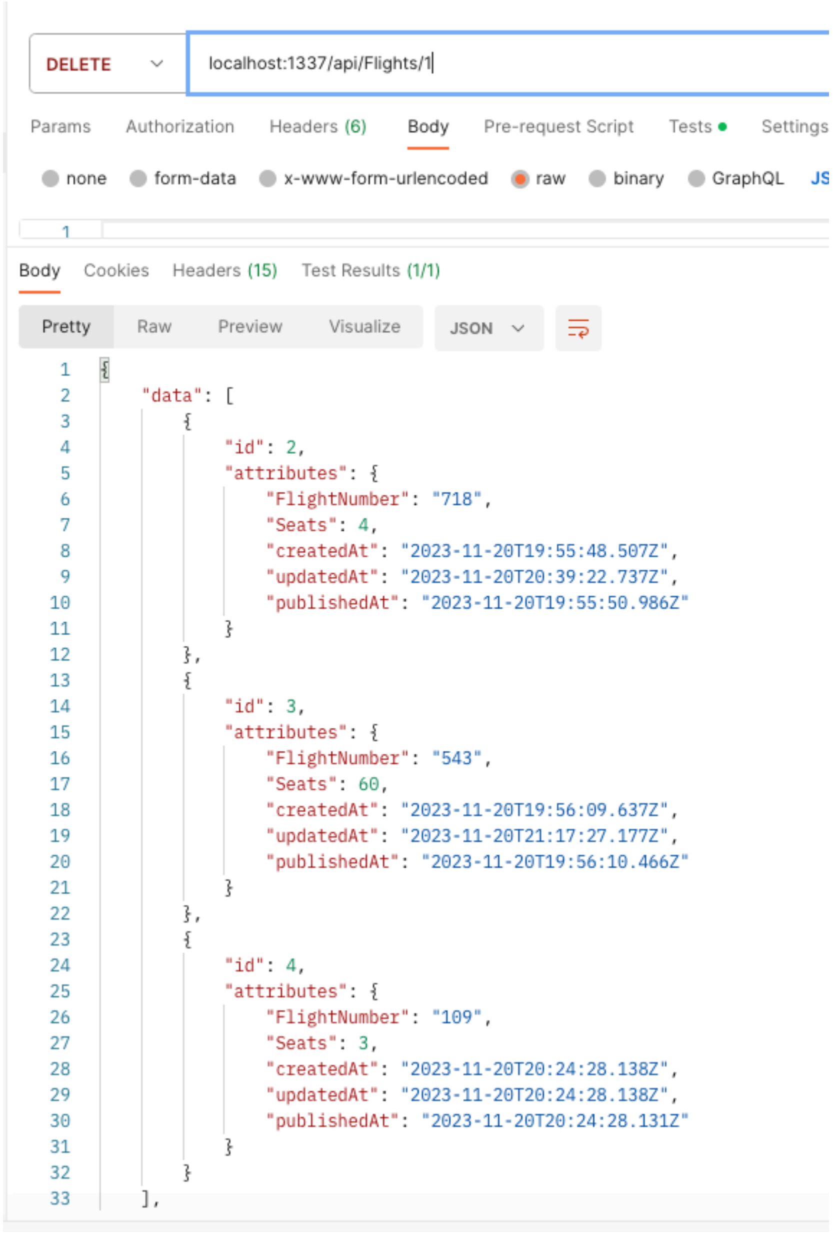
The next step was to send a DELETE request in Postman. Here, you can see all flight info for all the entries (you'll see the deleted entry further down):



localhost:1337/api/Flights

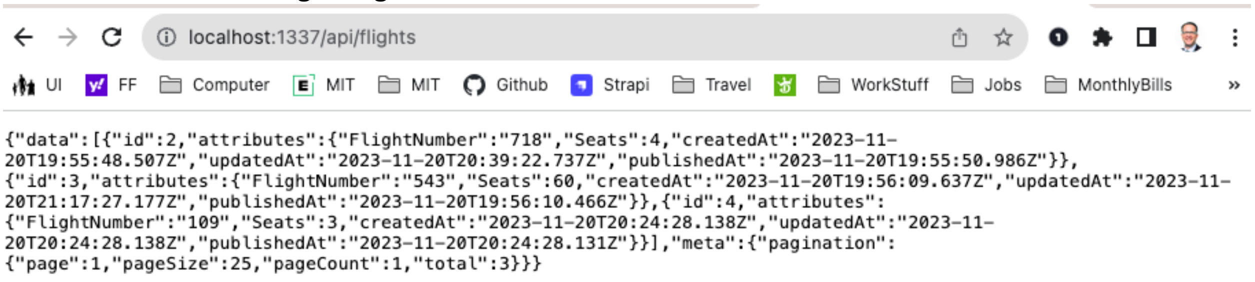
```
{
    "data": [
        {
            "id": 1,
            "attributes": {
                "FlightNumber": "123",
                "Seats": 2,
                "createdAt": "2023-11-20T19:55:16.107Z",
                "updatedAt": "2023-11-20T20:38:55.731Z",
                "publishedAt": "2023-11-20T19:55:27.504Z"
            }
        },
        {
            "id": 2,
            "attributes": {
                "FlightNumber": "718",
                "Seats": 4,
                "createdAt": "2023-11-20T19:55:48.507Z",
                "updatedAt": "2023-11-20T20:39:22.737Z",
                "publishedAt": "2023-11-20T19:55:50.986Z"
            }
        },
        {
            "id": 3,
            "attributes": {
                "FlightNumber": "543",
                "Seats": 60,
                "createdAt": "2023-11-20T19:56:09.637Z",
                "updatedAt": "2023-11-20T21:17:27.177Z",
                "publishedAt": "2023-11-20T19:56:10.466Z"
            }
        },
        {
```

And now you can see that flight with ID 1 is gone. (P.S. when I ran the DELETE API it returned the info for the flight I deleted).



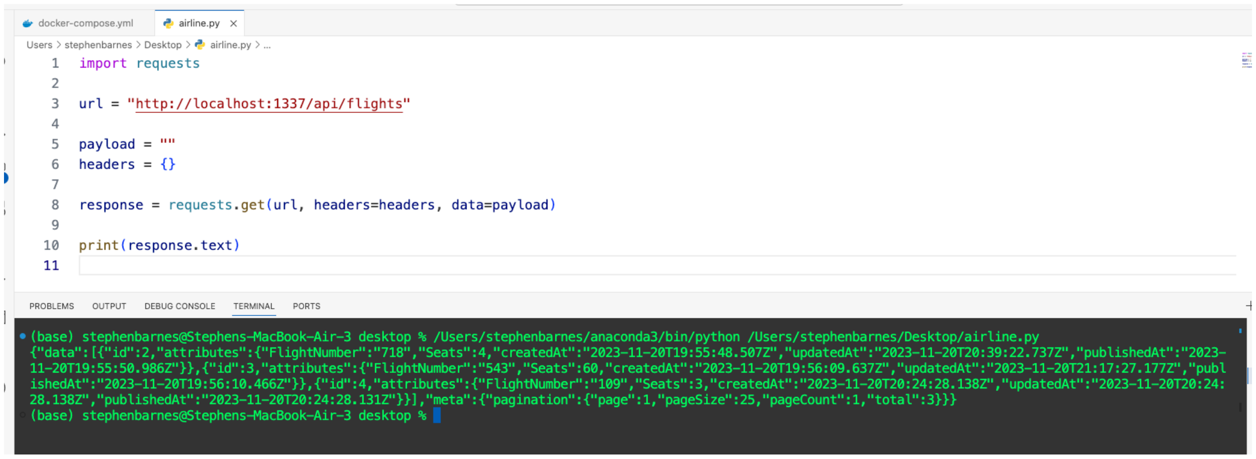
```
{
    "data": [
        {
            "id": 2,
            "attributes": {
                "FlightNumber": "718",
                "Seats": 4,
                "createdAt": "2023-11-20T19:55:48.507Z",
                "updatedAt": "2023-11-20T20:39:22.737Z",
                "publishedAt": "2023-11-20T19:55:50.986Z"
            }
        },
        {
            "id": 3,
            "attributes": {
                "FlightNumber": "543",
                "Seats": 60,
                "createdAt": "2023-11-20T19:56:09.637Z",
                "updatedAt": "2023-11-20T21:17:27.177Z",
                "publishedAt": "2023-11-20T19:56:10.466Z"
            }
        },
        {
            "id": 4,
            "attributes": {
                "FlightNumber": "109",
                "Seats": 3,
                "createdAt": "2023-11-20T20:24:28.138Z",
                "updatedAt": "2023-11-20T20:24:28.138Z",
                "publishedAt": "2023-11-20T20:24:28.131Z"
            }
        }
    ],
```

You can see that the flight is gone in the url too:



{"data":[{"id":2,"attributes":{"FlightNumber":"718","Seats":4,"createdAt":"2023-11-20T19:55:48.507Z","updatedAt":"2023-11-20T20:39:22.737Z","publishedAt":"2023-11-20T19:55:50.986Z"}},{"id":3,"attributes":{"FlightNumber":"543","Seats":60,"createdAt":"2023-11-20T19:56:09.637Z","updatedAt":"2023-11-20T21:17:27.177Z","publishedAt":"2023-11-20T19:56:10.466Z"}},{"id":4,"attributes":{"FlightNumber":"109","Seats":3,"createdAt":"2023-11-20T20:24:28.138Z","updatedAt":"2023-11-20T20:24:28.138Z","publishedAt":"2023-11-20T20:24:28.131Z"}}],"meta":{"pagination":{"page":1,"pageSize":25,"pageCount":1,"total":3}}}

The last step was to obtain the code to run a Python application to make a GET request in VS Code. This was done by clicking the ‘code’ icon on the right-hand panel within Postman and selecting the appropriate language (Python), and then clicking the copy icon for that language. Then, I created a new Python file from within VS Code and pasted the code from Postman.



```python
import requests

url = "http://localhost:1337/api/flights"

payload = ""
headers = {}

response = requests.get(url, headers=headers, data=payload)

print(response.text)

```

```
(base) stephenbarnes@Stephens-MacBook-Air-3 desktop % /Users/stephenbarnes/anaconda3/bin/python /Users/stephenbarnes/Desktop/airline.py
{"data":[{"id":2,"attributes":{"FlightNumber":"718","Seats":4,"createdAt":"2023-11-20T19:55:48.507Z","updatedAt":"2023-11-20T20:39:22.737Z","publishedAt":"2023-
11-20T19:55:50.986Z"}},{"id":3,"attributes":{"FlightNumber":"543","Seats":60,"createdAt":"2023-11-20T19:56:09.637Z","updatedAt":"2023-11-20T21:17:27.177Z","publ
ishedAt":"2023-11-20T19:56:10.466Z"}},{"id":4,"attributes":{"FlightNumber":"109","Seats":3,"createdAt":"2023-11-20T20:24:28.138Z","updatedAt":"2023-11-20T20:24:
28.138Z","publishedAt":"2023-11-20T20:24:28.131Z"}}],"meta":{"pagination":{"page":1,"pageSize":25,"pageCount":1,"total":3}}}
(base) stephenbarnes@Stephens-MacBook-Air-3 desktop %
```