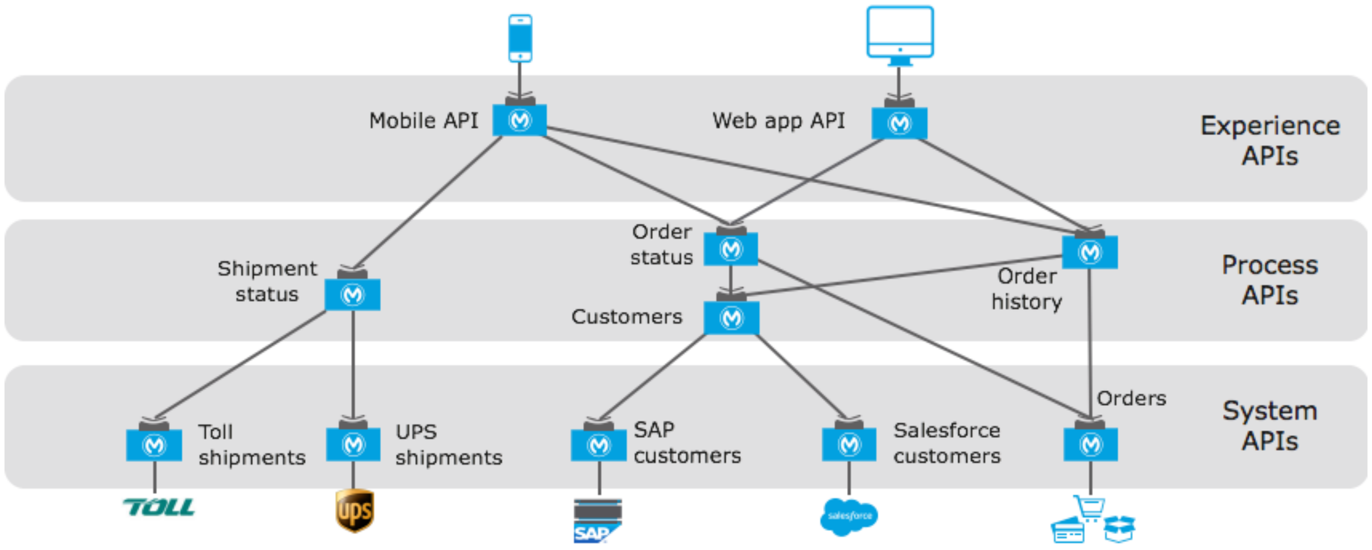
Module 1: Introducing Application Networks and API-Led Connectivity



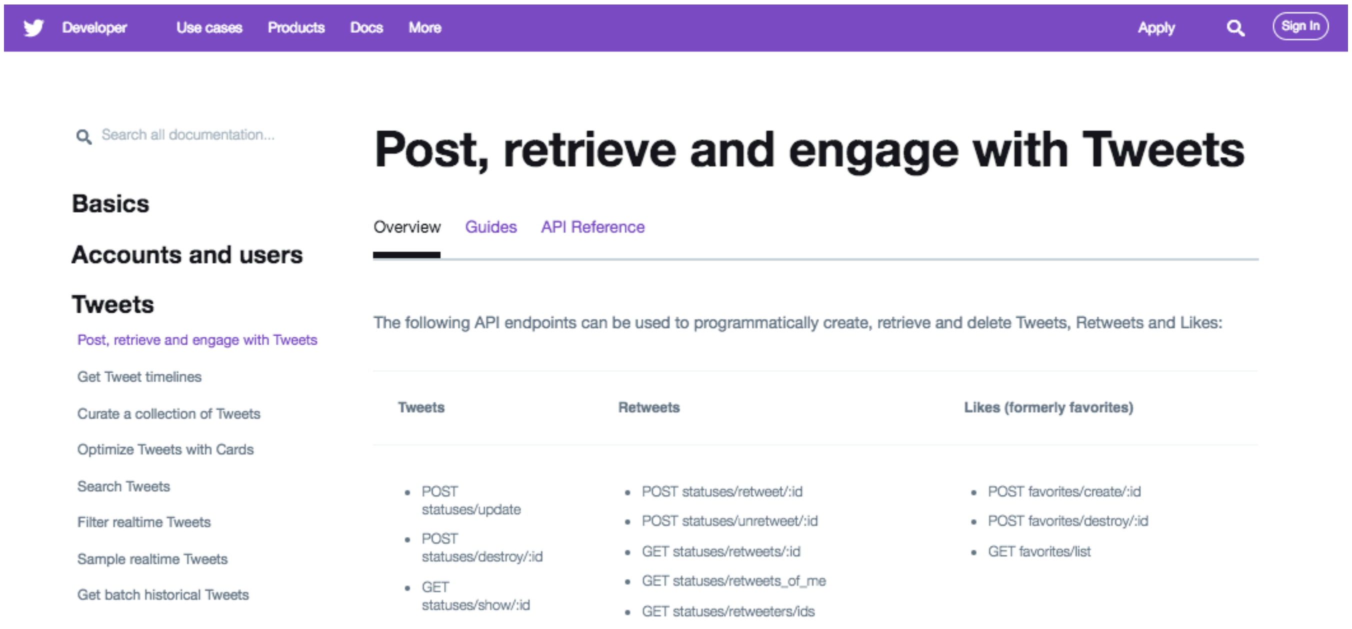
At the end of this module, you should be able to:

- Explain what an application network is and its benefits.
- Describe how to build an application network using API-led connectivity.
- Explain what web services and APIs are.
- Explore API directories and references.
- Make calls to secure and unsecured APIs.

Walkthrough 1-1: Explore an API directory and an API reference

In this walkthrough, you make calls to a RESTful API. You will:

- Browse the ProgrammableWeb API directory.
- Explore the API Reference for the Twitter API.



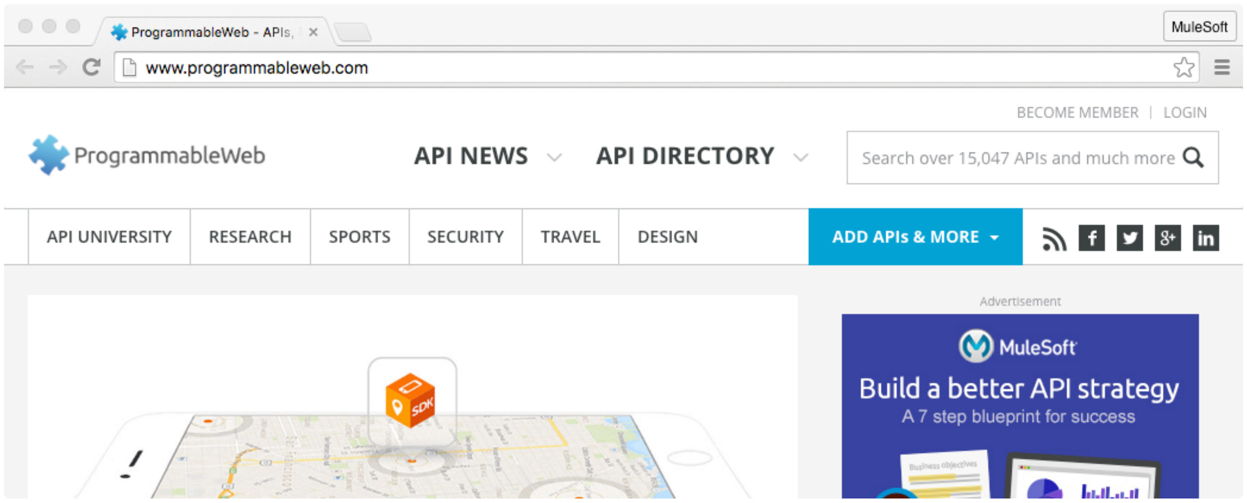
Post, retrieve and engage with Tweets

The following API endpoints can be used to programmatically create, retrieve and delete Tweets, Retweets and Likes:

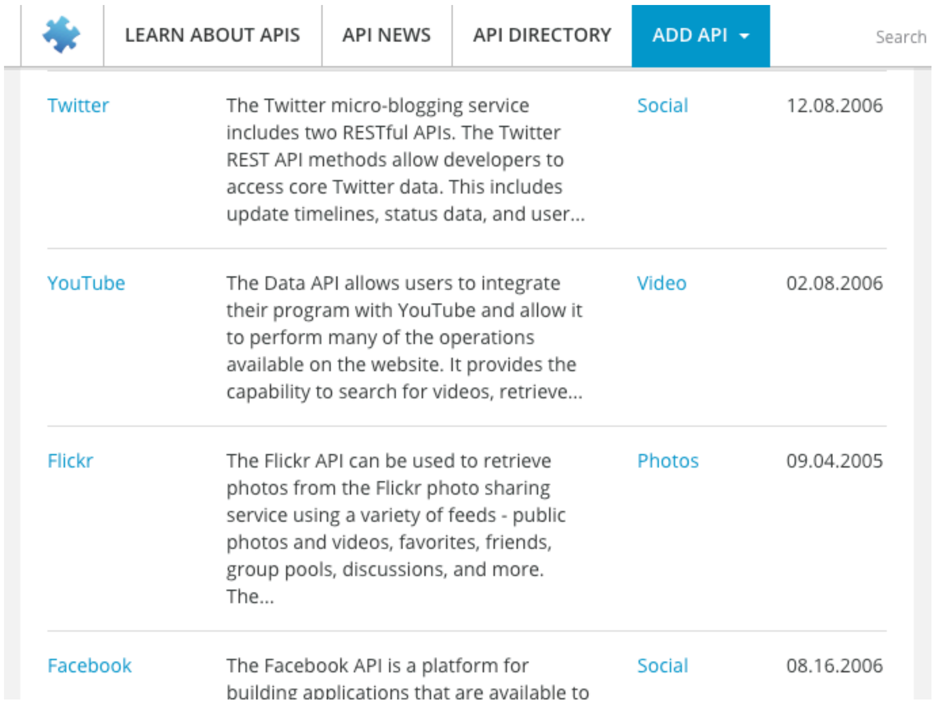
Tweets	Retweets	Likes (formerly favorites)
POST statuses/update	POST statuses/retweet/:id	POST favorites/create/:id
POST statuses/destroy/:id	POST statuses/unretweet/:id	POST favorites/destroy/:id
GET statuses/show/:id	GET statuses/retweets/:id	GET favorites/list
	GET statuses/retweets_of_me	
	GET statuses/retweeters/ids	

Browse the ProgrammableWeb API directory

1. In a web browser, navigate to http://www.programmableweb.com/.
2. Click the API directory link.



3. Browse the list of popular APIs.

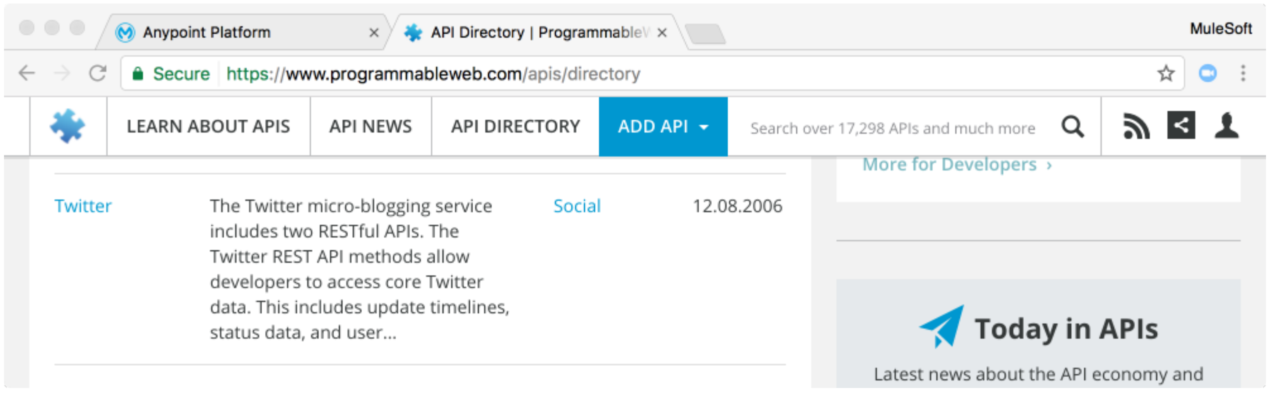


Twitter	The Twitter micro-blogging service includes two RESTful APIs. The Twitter REST API methods allow developers to access core Twitter data. This includes update timelines, status data, and user...	Social	12.08.2006
YouTube	The Data API allows users to integrate their program with YouTube and allow it to perform many of the operations available on the website. It provides the capability to search for videos, retrieve...	Video	02.08.2006
Flickr	The Flickr API can be used to retrieve photos from the Flickr photo sharing service using a variety of feeds - public photos and videos, favorites, friends, group pools, discussions, and more. The...	Photos	09.04.2005
Facebook	The Facebook API is a platform for building applications that are available to	Social	08.16.2006

Explore the API reference for the Twitter API

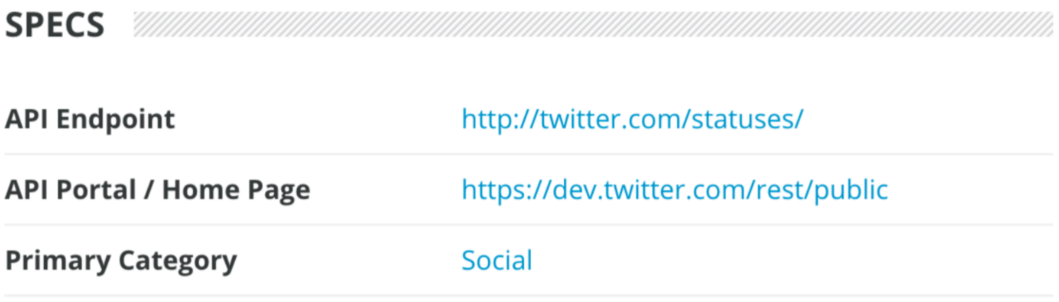
4. Scroll down and click the link for the Twitter API.

Note: If Twitter is no longer displayed on the main page, search for it.


Anypoint Platform
API Directory | ProgrammableW
MuleSoft
Secure
https://www.programmableweb.com/apis/directory
LEARN ABOUT APIS
API NEWS
API DIRECTORY
ADD API
Search over 17,298 APIs and much more
More for Developers ›
Today in APIs
Latest news about the API economy and

Twitter	The Twitter micro-blogging service includes two RESTful APIs. The Twitter REST API methods allow developers to access core Twitter data. This includes update timelines, status data, and user...	Social	12.08.2006

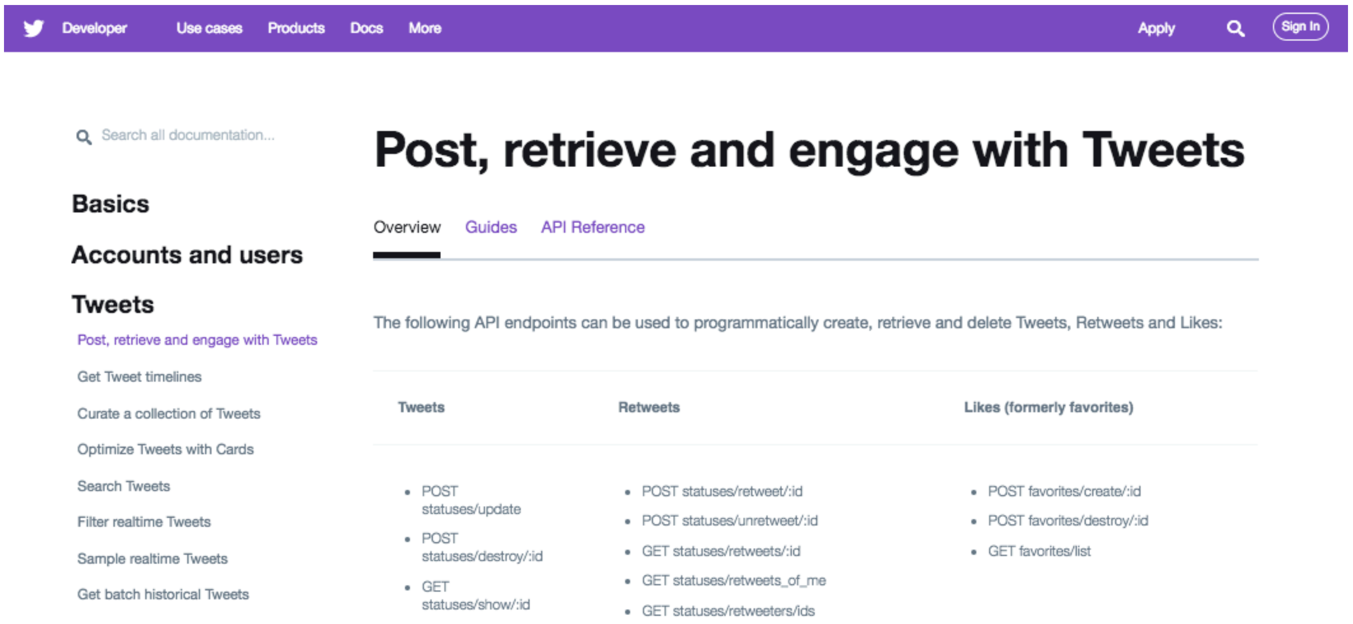
5. In the Specs section, click the API Portal / Home Page link.



SPECS

API Endpoint	http://twitter.com/statuses/
API Portal / Home Page	https://dev.twitter.com/rest/public
Primary Category	Social

6. In the new browser tab that opens, click Tweets in the left-side navigation.
7. In the left-side navigation, click Post, retrieve and engage with Tweets.



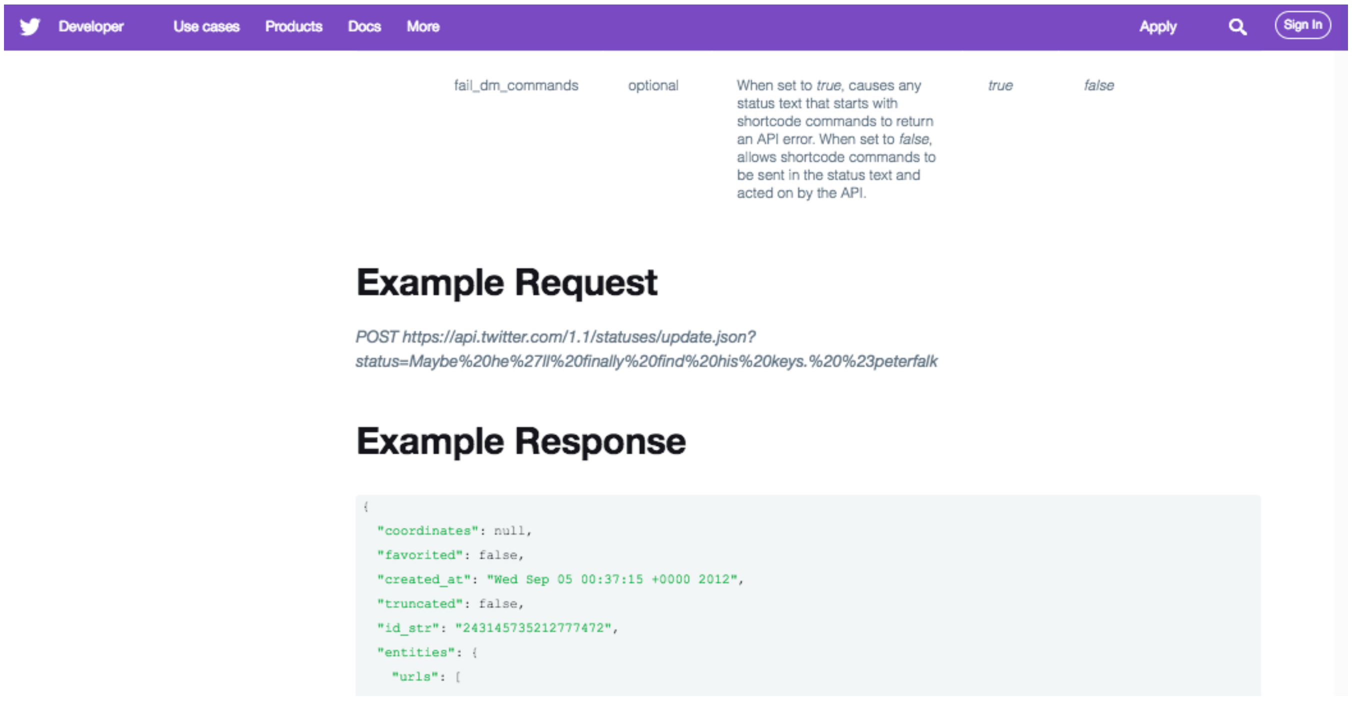
Post, retrieve and engage with Tweets

The following API endpoints can be used to programmatically create, retrieve and delete Tweets, Retweets and Likes:

Tweets	Retweets	Likes (formerly favorites)
• POST statuses/update	• POST statuses/retweet/:id	• POST favorites/create/:id
• POST statuses/destroy/:id	• POST statuses/unretweet/:id	• POST favorites/destroy/:id
• GET statuses/show/:id	• GET statuses/retweets/:id	• GET favorites/list
	• GET statuses/retweets_of_me	
	• GET statuses/retweeters/ids	

8. Browse the list of requests you can make to the API.
9. Select the API Reference tab beneath the title of the page.

10. Review the information for POST statuses/update including parameters, example request, and example response.



fail_dm_commands	optional	When set to *true*, causes any status text that starts with shortcode commands to return an API error. When set to *false*, allows shortcode commands to be sent in the status text and acted on by the API.	*true*	*false*

Example Request

POST https://api.twitter.com/1.1/statuses/update.json?status=Maybe%20he%27ll%20finally%20find%20his%20keys.%20%23peterfalk

Example Response

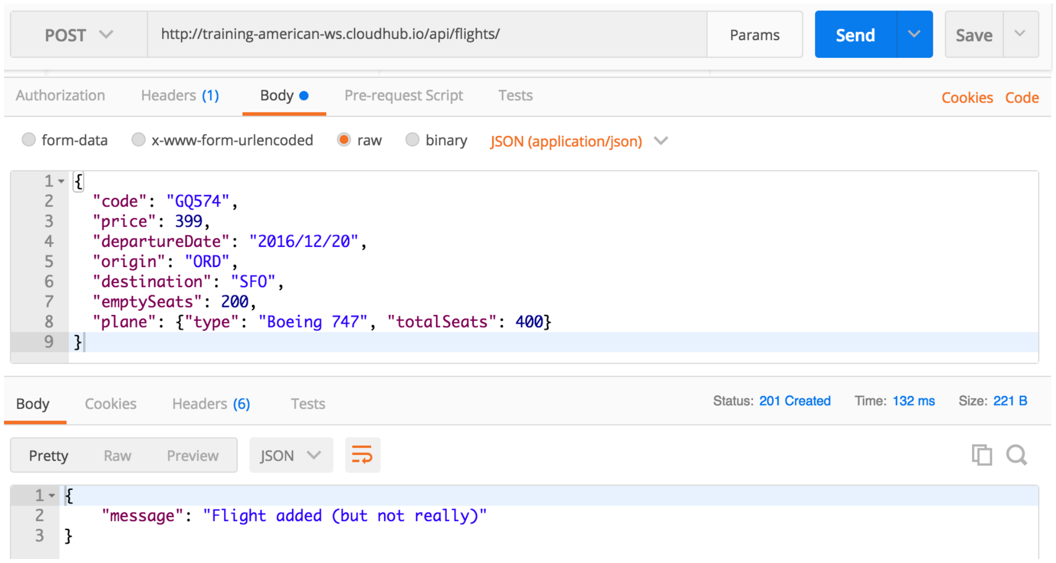
```
{
  "coordinates": null,
  "favorited": false,
  "created_at": "Wed Sep 05 00:37:15 +0000 2012",
  "truncated": false,
  "id_str": "243145735212777472",
  "entities": {
    "urls": [
```

11. Close the two browser tabs.

Walkthrough 1-2: Make calls to an API

In this walkthrough, you make calls to a RESTful API. You will:

- Use Postman to make calls to an unsecured API (an implementation).
- Make GET, DELETE, POST, and PUT calls.
- Use Postman to make calls to a secured API (an API proxy).



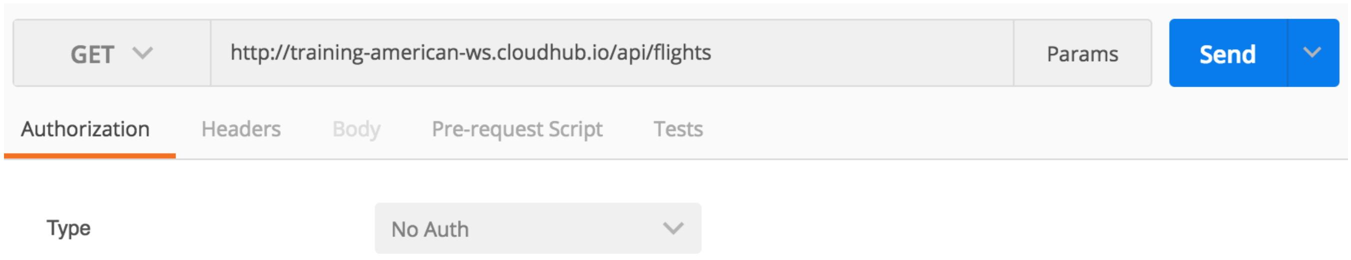
```
{
  "code": "GQ574",
  "price": 399,
  "departureDate": "2016/12/20",
  "origin": "ORD",
  "destination": "SFO",
  "emptySeats": 200,
  "plane": {"type": "Boeing 747", "totalSeats": 400}
}
```

Status: 201 Created Time: 132 ms Size: 221 B

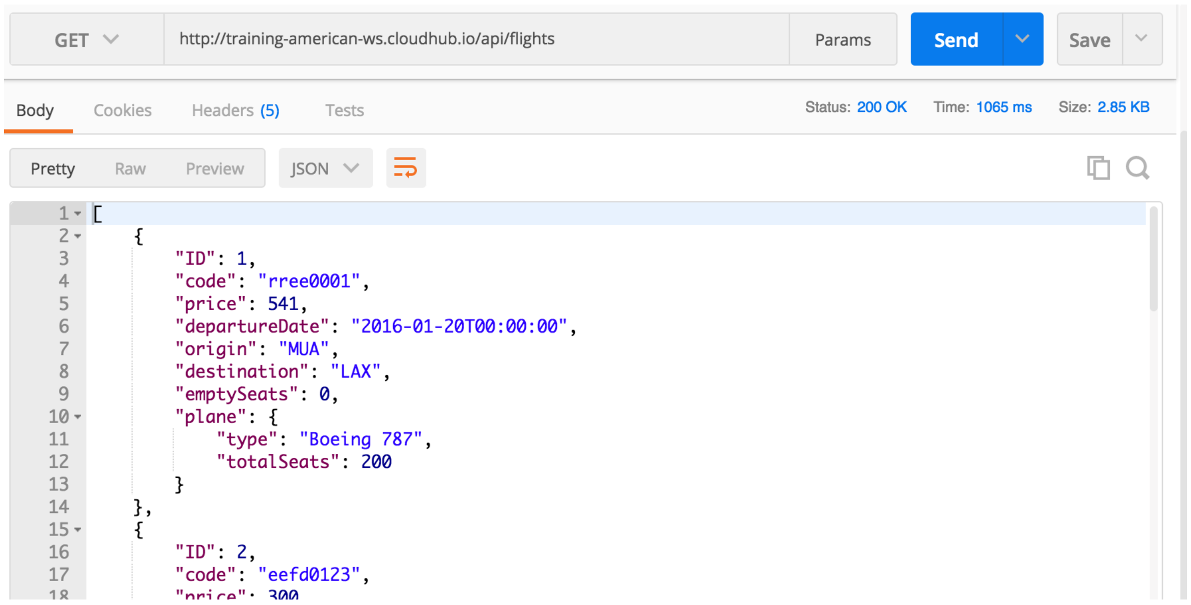
```
{
    "message": "Flight added (but not really)"
}
```

Make GET requests to retrieve data

1. Return to or open Postman.
2. Make sure the method is set to GET.
3. Return to the course snippets.txt file.
4. Copy the URL for the American Flights web service:
   http://training-american-ws.cloudhub.io/api/flights.
5. Return to Postman and paste the URL in the text box that says Enter request URL.



6. Click the Send button; you should get a response.
7. Locate and click the return HTTP status code of 200.
8. Review the response body containing flights to SFO, LAX, and CLE.

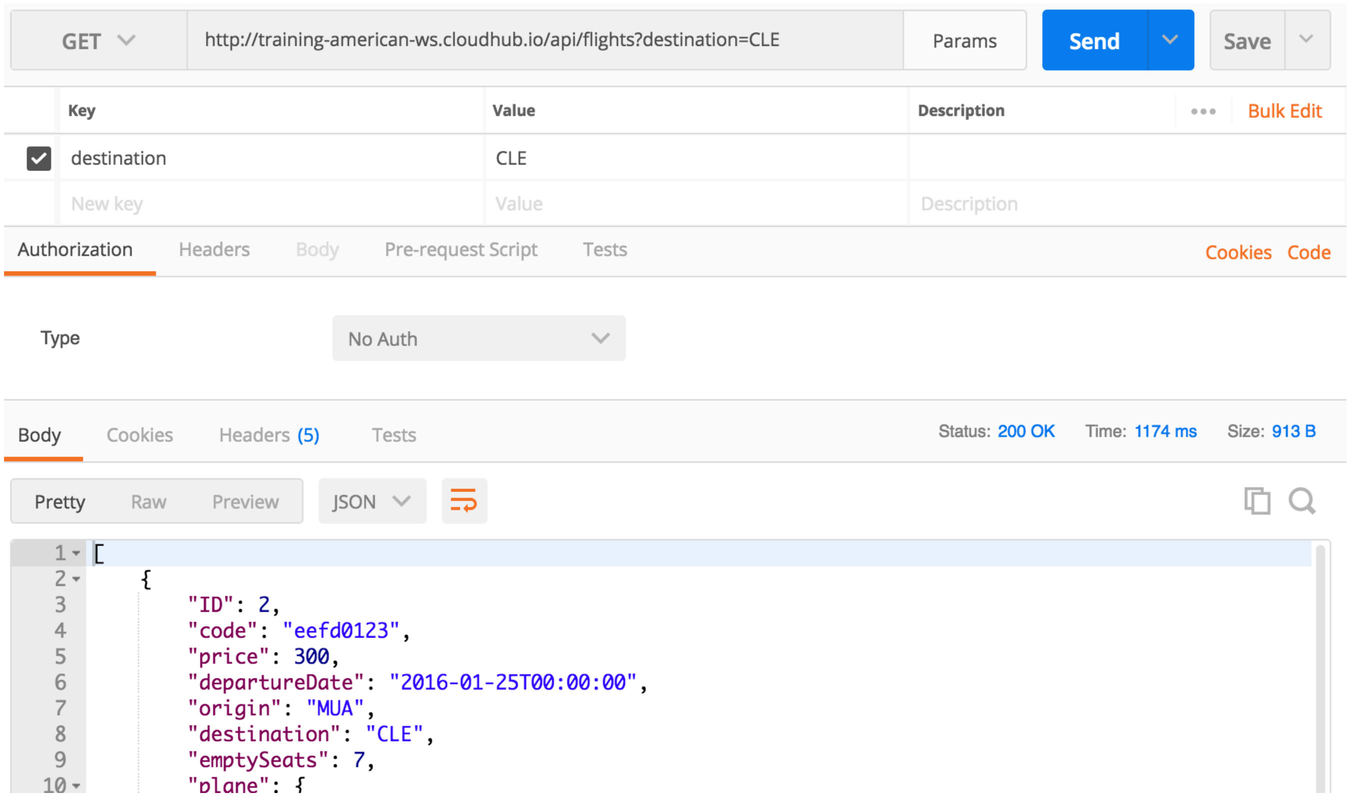


http://training-american-ws.cloudhub.io/api/flights

Status: 200 OK Time: 1065 ms Size: 2.85 KB

```
[
    {
        "ID": 1,
        "code": "rree0001",
        "price": 541,
        "departureDate": "2016-01-20T00:00:00",
        "origin": "MUA",
        "destination": "LAX",
        "emptySeats": 0,
        "plane": {
            "type": "Boeing 787",
            "totalSeats": 200
        }
    },
    {
        "ID": 2,
        "code": "eefd0123",
        "price": 300
```

9. Click the Params button next to the URL.
10. In the area that appears, set the key to destination and the value to CLE.
11. Click the Send button; you should get just flights to CLE returned.



http://training-american-ws.cloudhub.io/api/flights?destination=CLE

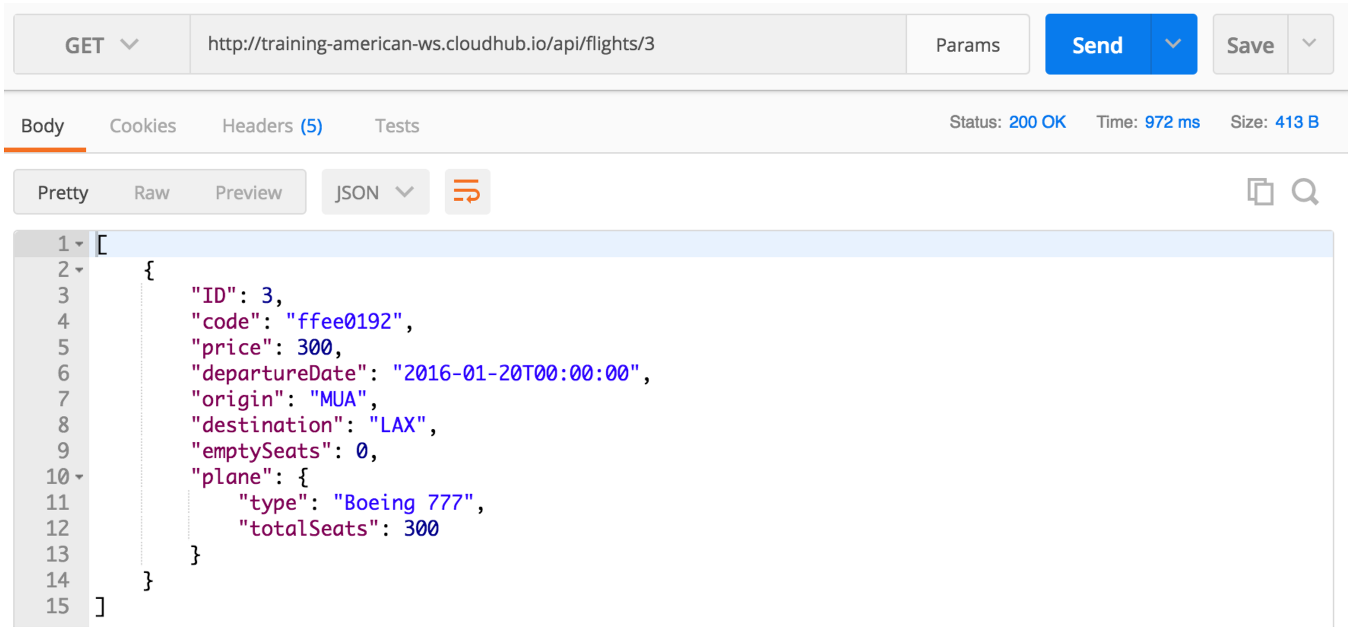
	Key	Value	Description
✓	destination	CLE	
	New key	Value	Description

Type: No Auth

Status: 200 OK Time: 1174 ms Size: 913 B

```
[
    {
        "ID": 2,
        "code": "eefd0123",
        "price": 300,
        "departureDate": "2016-01-25T00:00:00",
        "origin": "MUA",
        "destination": "CLE",
        "emptySeats": 7,
        "plane": {
```

12. Click the X next to the parameter to delete it.
13. Change the request URL to use a uri parameter to retrieve the flight with an ID of 3: http://training-american-ws.cloudhub.io/api/flights/3
14. Click the Send button; you should see only the flight with that ID returned.

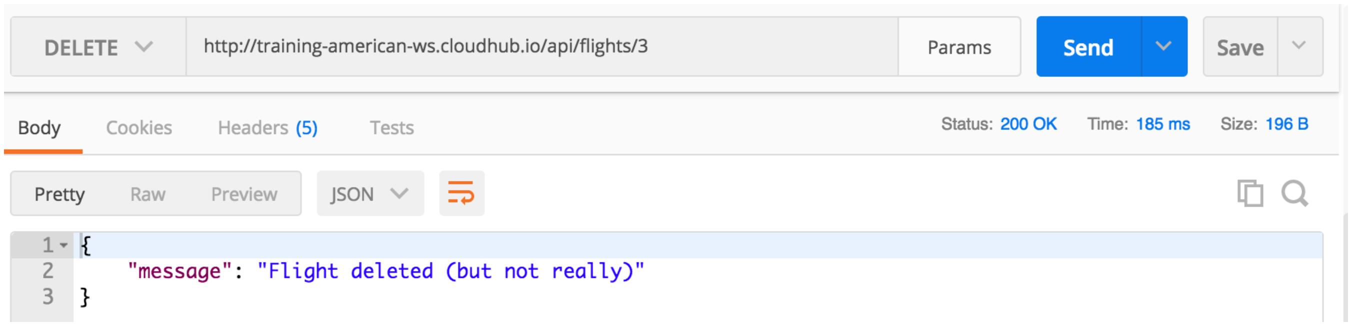


```
[
    {
        "ID": 3,
        "code": "ffee0192",
        "price": 300,
        "departureDate": "2016-01-20T00:00:00",
        "origin": "MUA",
        "destination": "LAX",
        "emptySeats": 0,
        "plane": {
            "type": "Boeing 777",
            "totalSeats": 300
        }
    }
]
```

Make DELETE requests to delete data

15. Change the method to DELETE.
16. Click the Send button; you should see a 200 response with a message that the Flight was deleted (but not really).

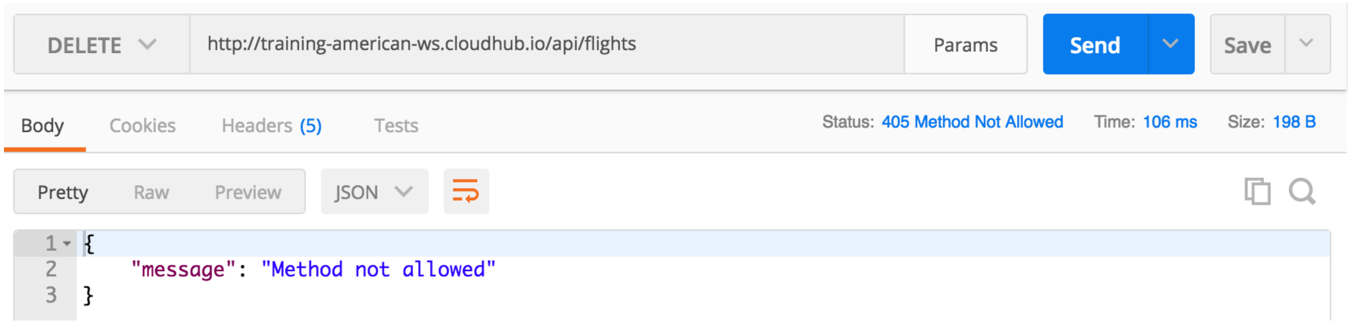
    Note: The database is not actually modified so that its data integrity can be retained for class.



```
{
    "message": "Flight deleted (but not really)"
}
```

17. Remove the URI parameter from the request: http://training-american-ws.cloudhub.io/api/flights.

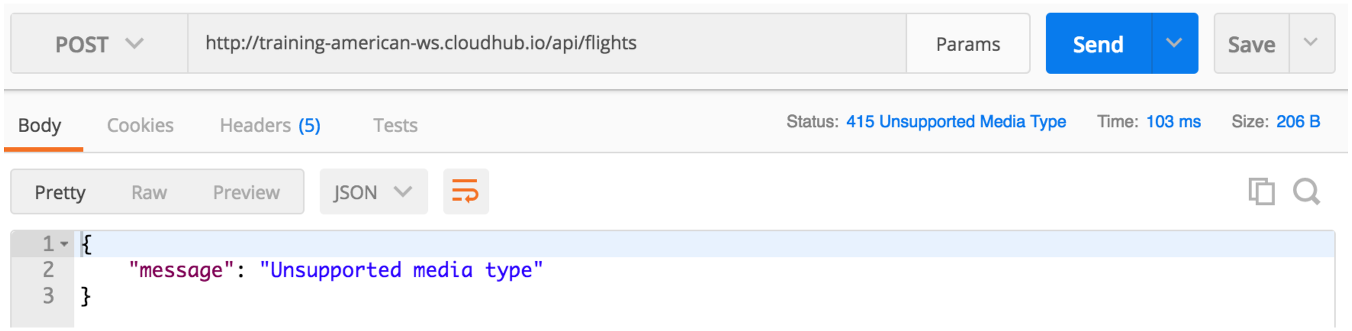
18. Click the Send button; you should get a 405 response with a message of method not allowed.



```
{
    "message": "Method not allowed"
}
```

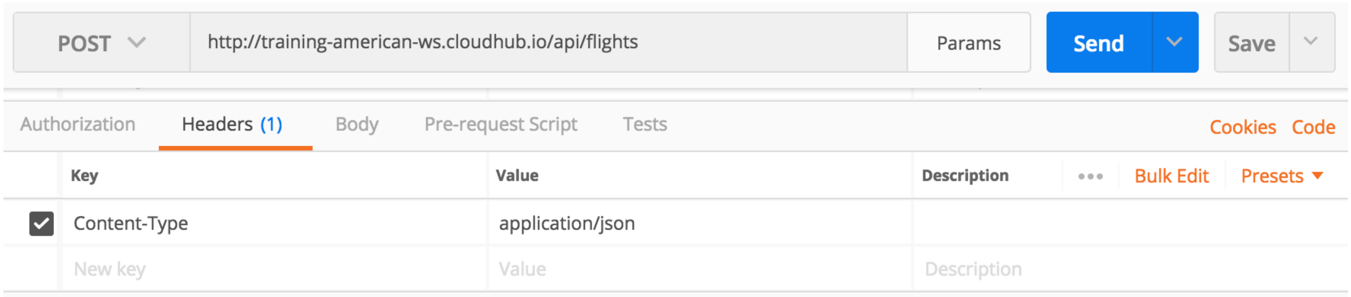
Make a POST request to add data

19. Change the method to POST.
20. Click the Send button; you should get a 415 response with a message of unsupported media type.



```
{
    "message": "Unsupported media type"
}
```

21. Click the Headers link under the request URL.
22. Click in the Headers key field, type C, and then select Content-Type.
23. Click in the Value field, type A, and then select application/json.

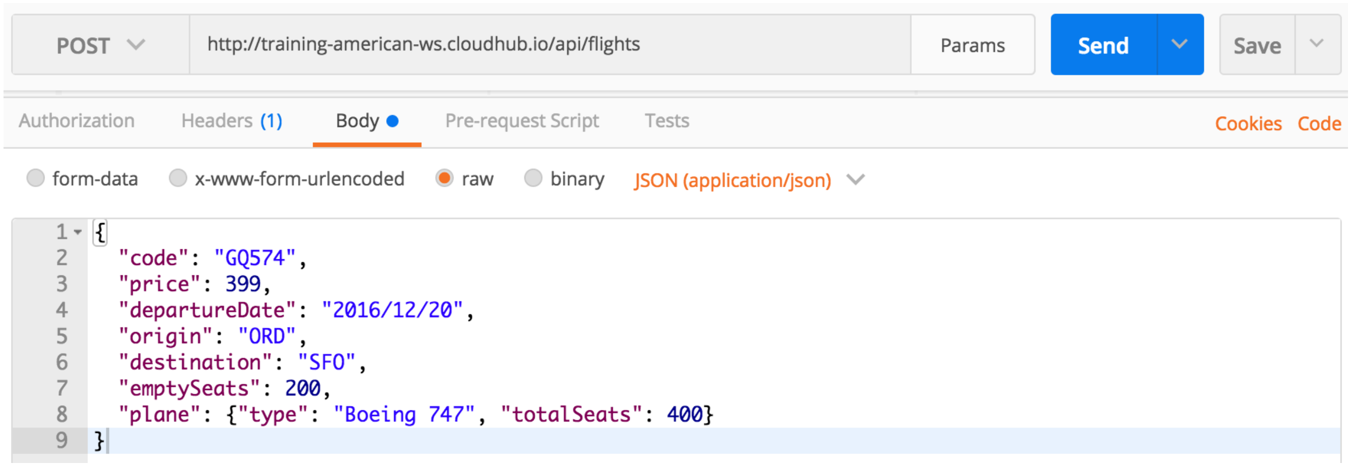


	Key	Value	Description
✓	Content-Type	application/json	
	New key	Value	Description

24. Click the Body link under the request URL.
25. Select the raw radio button.
26. Return to the course snippets.txt file and copy the value for American Flights API post body.

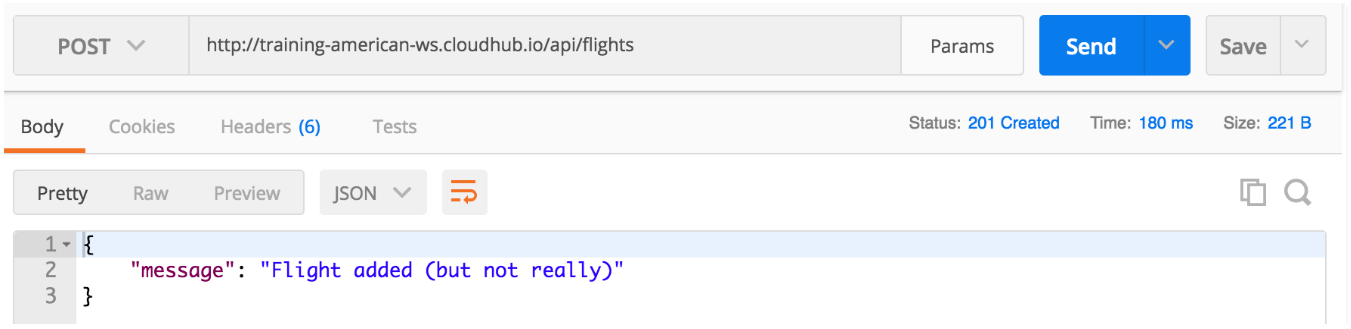
MuleSoft

27. Return to Postman and paste the code in the body text area.



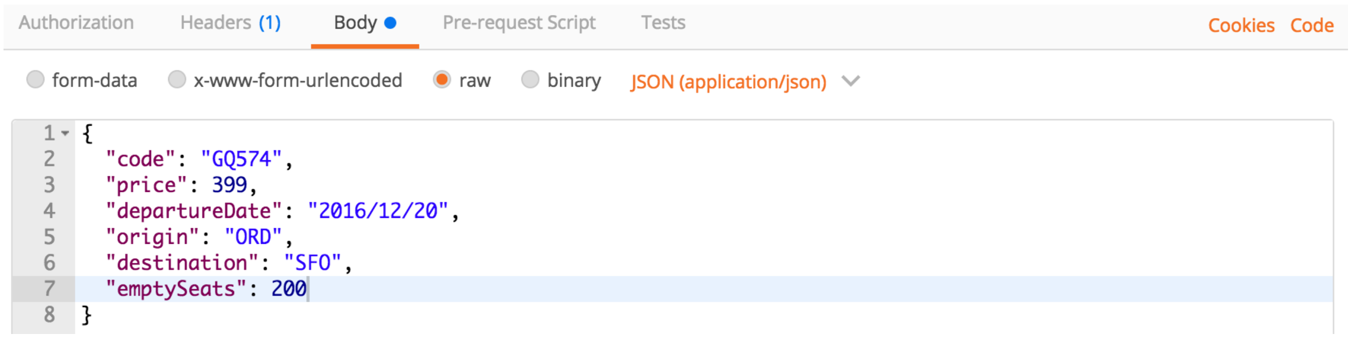
```
{
  "code": "GQ574",
  "price": 399,
  "departureDate": "2016/12/20",
  "origin": "ORD",
  "destination": "SFO",
  "emptySeats": 200,
  "plane": {"type": "Boeing 747", "totalSeats": 400}
}
```

28. Click the Send button; you should see a 201 response with the message Flight added (but not really).



```
{
    "message": "Flight added (but not really)"
}
```

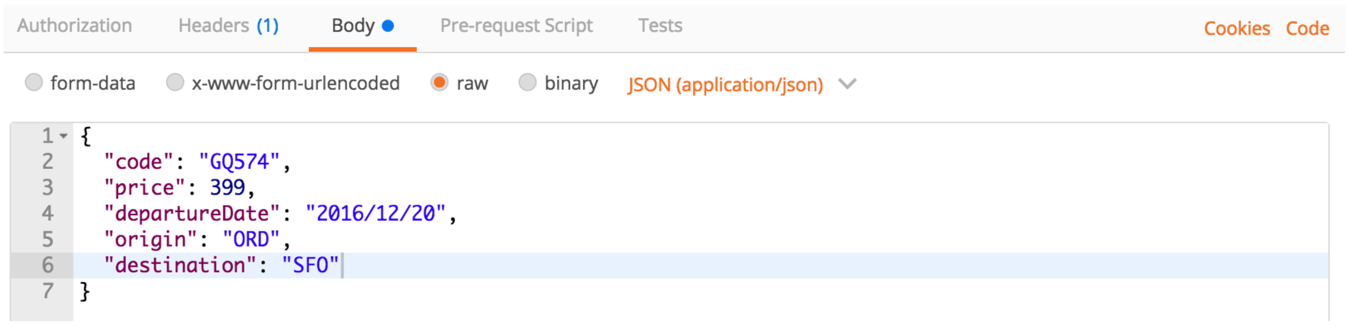
29. Return to the request body and remove the plane field and value from the request body.
30. Remove the comma after the emptySeats key/value pair.



```
{
  "code": "GQ574",
  "price": 399,
  "departureDate": "2016/12/20",
  "origin": "ORD",
  "destination": "SFO",
  "emptySeats": 200
}
```

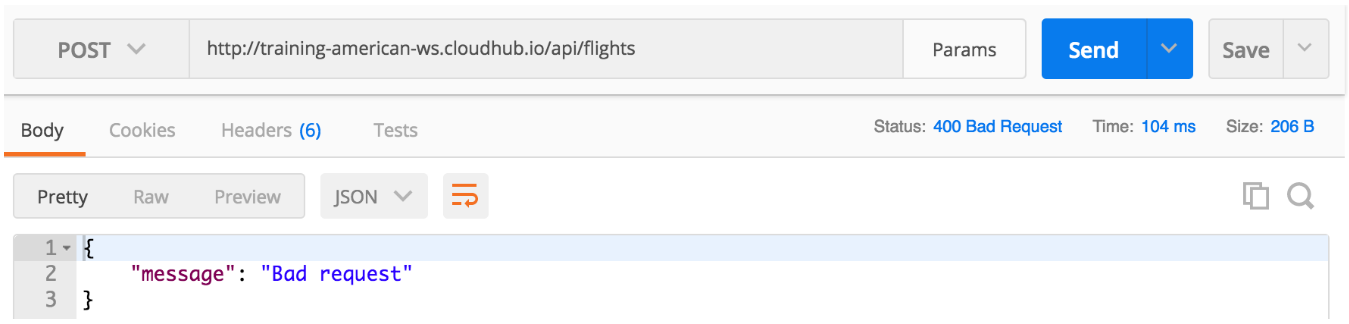
31. Send the request; the message should still post successfully.

32. In the request body, remove the emptySeats key/value pair.
33. Delete the comma after the destination key/value pair.



```
{
  "code": "GQ574",
  "price": 399,
  "departureDate": "2016/12/20",
  "origin": "ORD",
  "destination": "SFO"
}
```

34. Send the request; you should see a 400 response with the message Bad request.

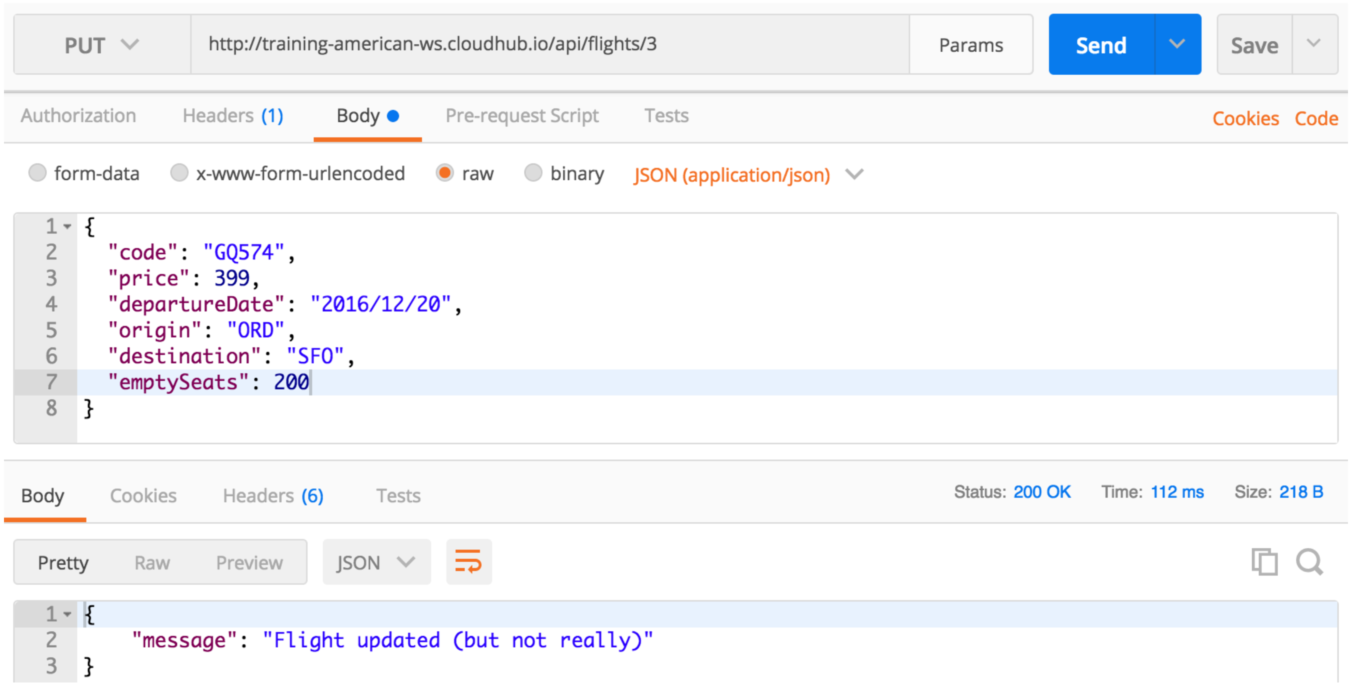


```
{
    "message": "Bad request"
}
```

Make a PUT request to update data

35. Change the method to PUT.
36. Add a flight ID to the URL to modify a particular flight.
37. Click the Send button; you should get a bad request message.
38. In the request body field, press Cmd+Z or Ctrl+Z so the emptySeats field is added back.

39. Send the request; you should see the response Flight updated (but not really).



```
http://training-american-ws.cloudhub.io/api/flights/3
```

```
{
  "code": "GQ574",
  "price": 399,
  "departureDate": "2016/12/20",
  "origin": "ORD",
  "destination": "SFO",
  "emptySeats": 200
}
```

Status: 200 OK Time: 112 ms Size: 218 B

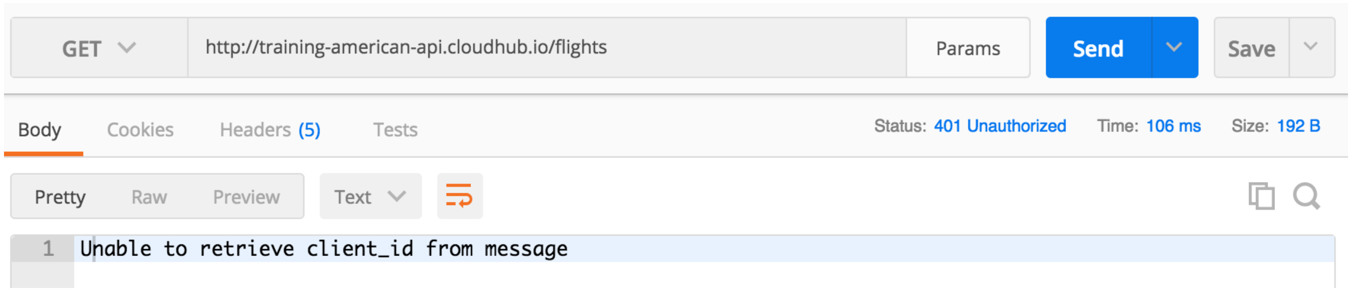
```
{
    "message": "Flight updated (but not really)"
}
```

Make a request to a secured API

40. Change the method to GET.
41. Change the request URL to http://training-american-api.cloudhub.io/flights/3.

    Note: The -ws in the URL has been changed to -api and the /api removed.

42. Click the Send button; you should get a message about a missing client_id.



```
http://training-american-api.cloudhub.io/flights
```

Status: 401 Unauthorized Time: 106 ms Size: 192 B

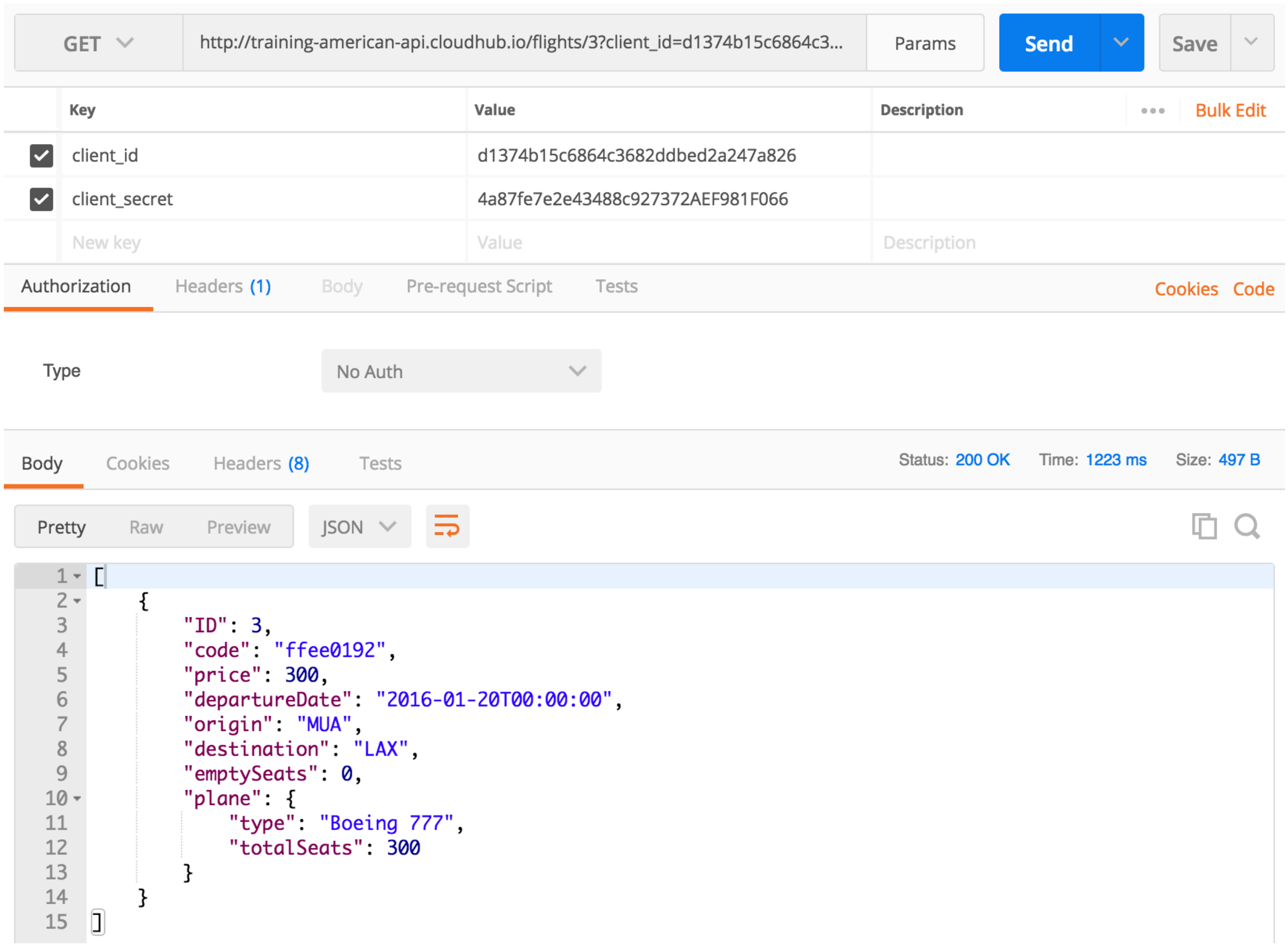
```
Unable to retrieve client_id from message
```

43. Return to the course snippets.txt file and copy the value for the American Flights API client_id.
44. Return to Postman and add a request parameter called client_id.
45. Set client_id to the value you copied from the snippets.txt file.
46. Return to the course snippets.txt file and copy the value for the American Flights API client_secret.
47. Return to Postman and add a request parameter called client_secret.

MuleSoft

48. Set client_secret to the value you copied from the snippets.txt file.
49. Click the Send button; you should get data for flight 3 again.

Note: The API service level agreement (SLA) for the application with this client ID and secret has been set to allow three API calls per minute.



GET
http://training-american-api.cloudhub.io/flights/3?client_id=d1374b15c6864c3...
Params
Send
Save
Bulk Edit

Key	Value	Description
✓ client_id	d1374b15c6864c3682ddbed2a247a826	
✓ client_secret	4a87fe7e2e43488c927372AEF981F066	
New key	Value	Description

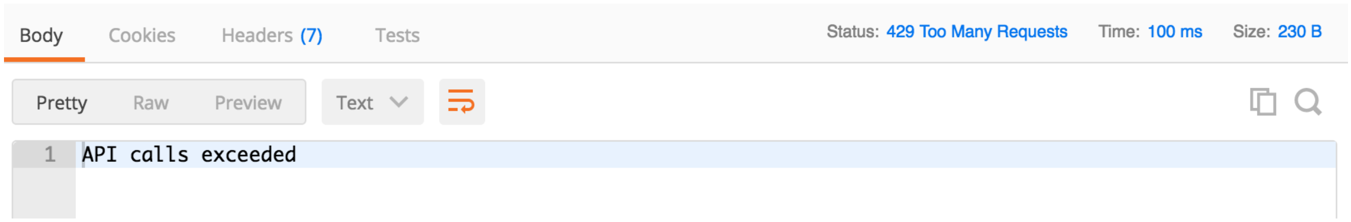
Type: No Auth

Status: 200 OK Time: 1223 ms Size: 497 B

```
[
    {
        "ID": 3,
        "code": "ffee0192",
        "price": 300,
        "departureDate": "2016-01-20T00:00:00",
        "origin": "MUA",
        "destination": "LAX",
        "emptySeats": 0,
        "plane": {
            "type": "Boeing 777",
            "totalSeats": 300
        }
    }
]
```

50. Click the Send button three more times; you should get a 429 response and an API calls exceeded message.

Note: The API service level agreement (SLA) for the application with this client ID and secret has been set to allow three API calls per minute.



Status: 429 Too Many Requests Time: 100 ms Size: 230 B

```
API calls exceeded
```