

Walkthrough 8-2: Consume a RESTful web service

In this walkthrough, you consume the United RESTful web service that does not have an API in Exchange. You will:

- Create a new flow to call the United RESTful web service.
- Use the HTTP Request operation to call a RESTful web service.
- Dynamically set a URI parameter for a web service call.
- Add metadata for an HTTP Request operation's response.

The screenshot displays the MuleSoft Studio interface. On the left, a flow diagram for 'getUnitedFlights' shows a sequence: a Listener (GET /united) connects to a Flow Reference (setCode), which then connects to a Request (Get flights). Below the flow is an 'Error handling' section. On the right, the 'Get flights' configuration pane is open, showing the 'Request' tab. The 'Method' is set to 'GET (Default)', the 'Path' is '/united/flights/{dest}', and the 'URL' is empty. The 'URI Parameters' tab is also visible, showing a parameter 'dest' with the value 'vars.code'. The 'Output' tab on the far right shows the 'Mule Message' payload, which is an array of objects. The first object in the array has the following fields: 'airlineName' (String), 'price' (Number), 'departureDate' (String), 'planeType' (String), 'origin' (String), 'code' (String), 'emptySeats' (Number), and 'destination' (String).

Make a request to the United web service

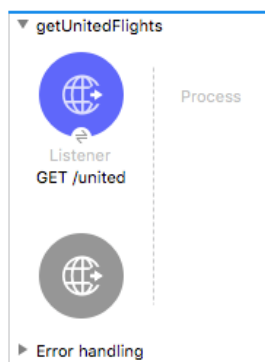
1. Return to the course snippets.txt file.
2. Locate and copy the United RESTful web service URL.
3. In Advanced REST Client, make a new tab and make a GET request to this URL.
4. Review the structure of the JSON response.

The screenshot shows the Advanced REST Client interface. The 'Method' is set to 'GET' and the 'Request URL' is 'http://mu.learn.mulesoft.com/united/flights'. A 'SEND' button is visible. Below the URL bar, the 'Parameters' section is expanded, showing a status of '200 OK' and a response time of '278.45 ms'. The 'DETAILS' button is also visible. The JSON response is displayed in a code editor, showing an array of flight objects. The first object in the array has the following fields: 'code' (ER38sd), 'price' (400), 'origin' (MUA), 'destination' (SFO), 'departureDate' (2015/03/20), 'airlineName' (United), 'planeType' (Boeing 737), and 'emptySeats' (0).

5. Look at the destination values; you should see SFO, LAX, CLE, PDX, and PDF.
6. In the URL field, add the destination CLE as a URI parameter:
<http://mu.learn.mulesoft.com/united/flights/CLE>.
7. Send the request; you should now only see flights to CLE.

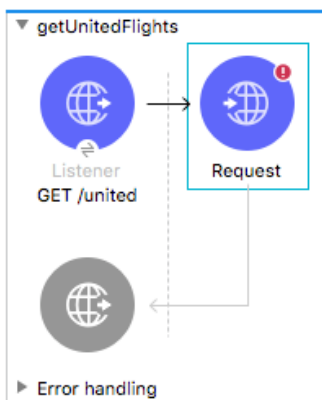
Add a new flow with an HTTP Listener operation

8. Return to implementation.xml in Anypoint Studio.
9. Drag out an HTTP Listener from the Mule Palette and drop it at the bottom of the canvas.
10. Change the name of the flow to getUnitedFlights.
11. In the Listener properties view, set the display name to GET /united.
12. Set the connector configuration to the existing HTTP_Listener_config.
13. Set the path to /united.
14. Set the allowed methods to GET.



Add an HTTP Request operation

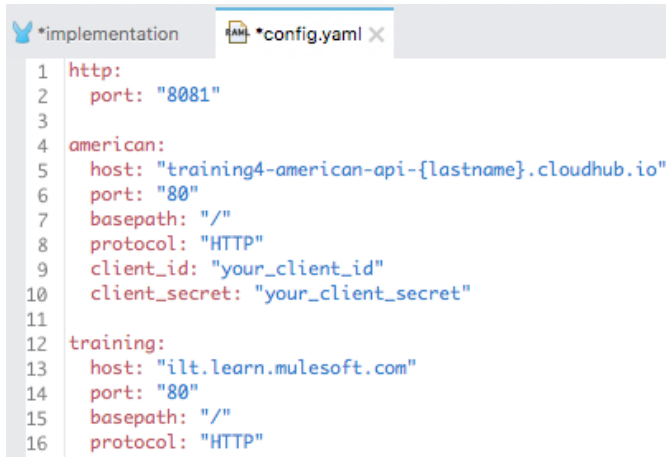
15. Drag out an HTTP Request from the Mule Palette and drop it into the process section of getUnitedFlights.



16. In the Request properties view, set the display name to Get flights.

Create an HTTP Request configuration

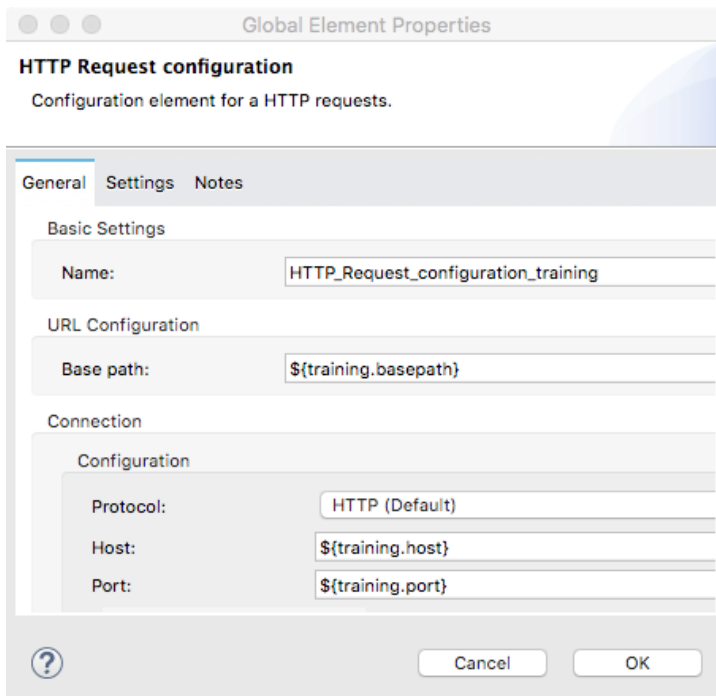
17. Return to the course snippets.txt file and copy the text for the Training web service properties.
18. Return to config.yaml and paste the code at the end of the file.



```
1 http:
2   port: "8081"
3
4 american:
5   host: "training4-american-api-{lastname}.cloudhub.io"
6   port: "80"
7   basepath: "/"
8   protocol: "HTTP"
9   client_id: "your_client_id"
10  client_secret: "your_client_secret"
11
12 training:
13   host: "ilt.learn.mulesoft.com"
14   port: "80"
15   basepath: "/"
16   protocol: "HTTP"
```

19. Return to the Global Elements view in global.xml.
20. Create a new HTTP Request configuration with the following values:

- Name: HTTP_Request_config_training
- Base Path: \${training.basepath}
- Host: \${training.host}
- Port: \${training.port}



Global Element Properties

HTTP Request configuration
Configuration element for a HTTP requests.

General Settings Notes

Basic Settings

Name: HTTP_Request_configuration_training

URL Configuration

Base path: \${training.basepath}

Connection

Configuration

Protocol: HTTP (Default)

Host: \${training.host}

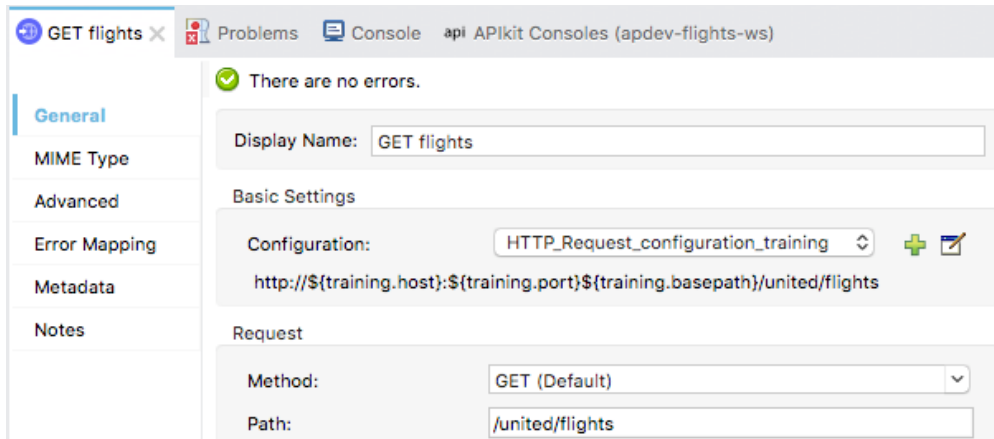
Port: \${training.port}

Cancel OK

21. Click OK.

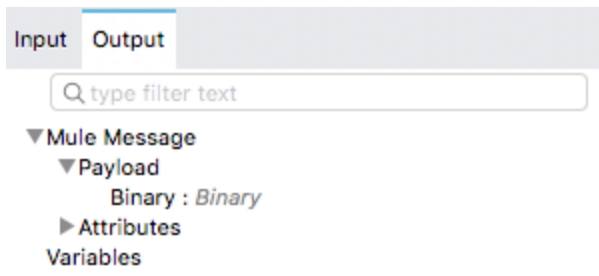
Configure the HTTP Request operation

22. Return to implementation.xml.
23. Navigate to the Get flights Request properties view in getUnitedFlights.
24. Set the configuration to the existing HTTP_Request_configuration_training.
25. Leave the method set to GET.
26. Set the path to /united/flights.



Review metadata associated with the United Get flights operation response

27. Select the Output tab in the DataSense Explorer and expand Payload; you should see no metadata for the payload.



Test the application

28. Save the files to redeploy the project.
29. In Advanced REST Client, return to the tab with the localhost request.

30. Change the URL to make a request to <http://localhost:8081/united>; you should get JSON flight data for all destinations returned.

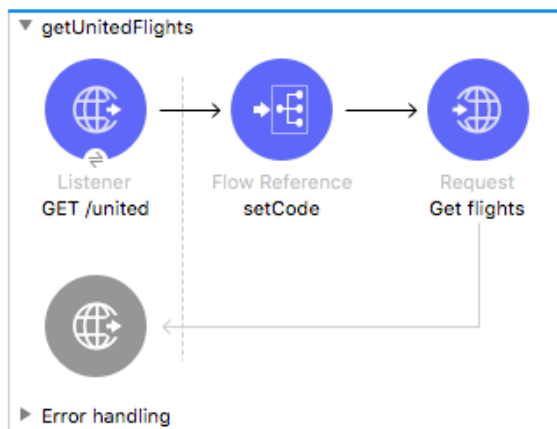
Method: GET, Request URL: <http://localhost:8081/united>, Status: 200 OK, 446.93 ms.

```
{
  "-flights": [Array[12]
    -0: {
      "code": "ER38sd",
      "price": 400,
      "origin": "MUA",
      "destination": "SFO",
      "departureDate": "2015/03/20",
      "planeType": "Boeing 737",
      "airlineName": "United",
      "emptySeats": 0
    },
    -1: {
      "code": "ER45if",
      "price": 345.99
    }
  ]
}
```

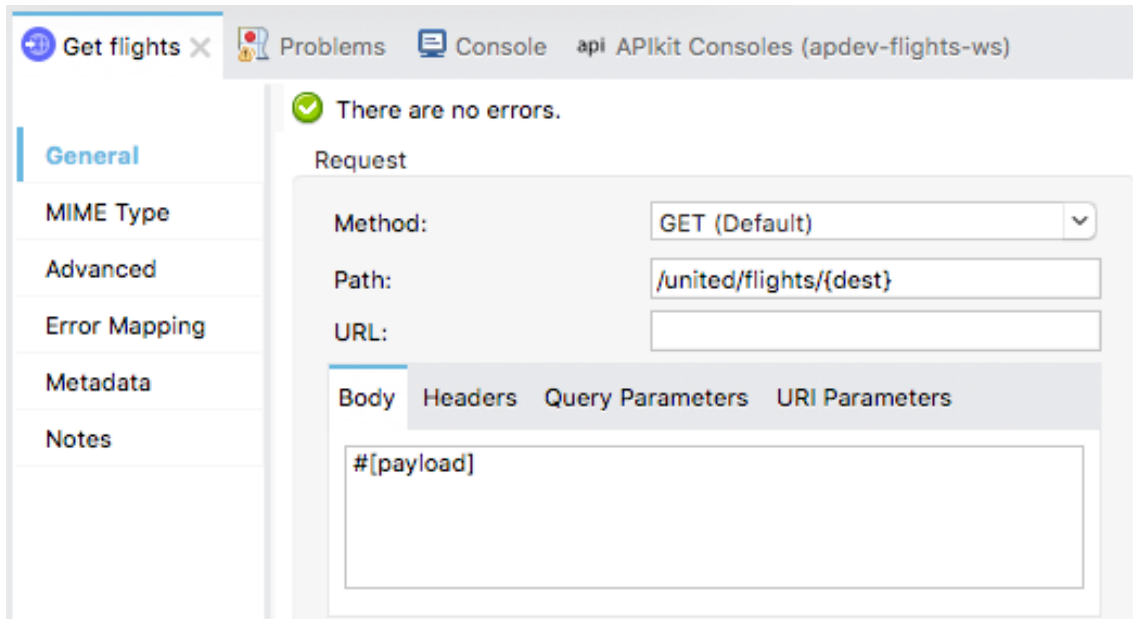
31. Add a query parameter named code and set it to one of the destination airport code values: <http://localhost:8081/united?code=CLE>; you should still get flights to all destinations.

Add a URI parameter to the web service call

32. Return to implementation.xml in Anypoint Studio.
33. Drag a Flow Reference component from the Mule Palette and drop it after the GET /united Listener in getUnitedFlights.
34. In the Flow Reference properties view, set the flow name to setCode.

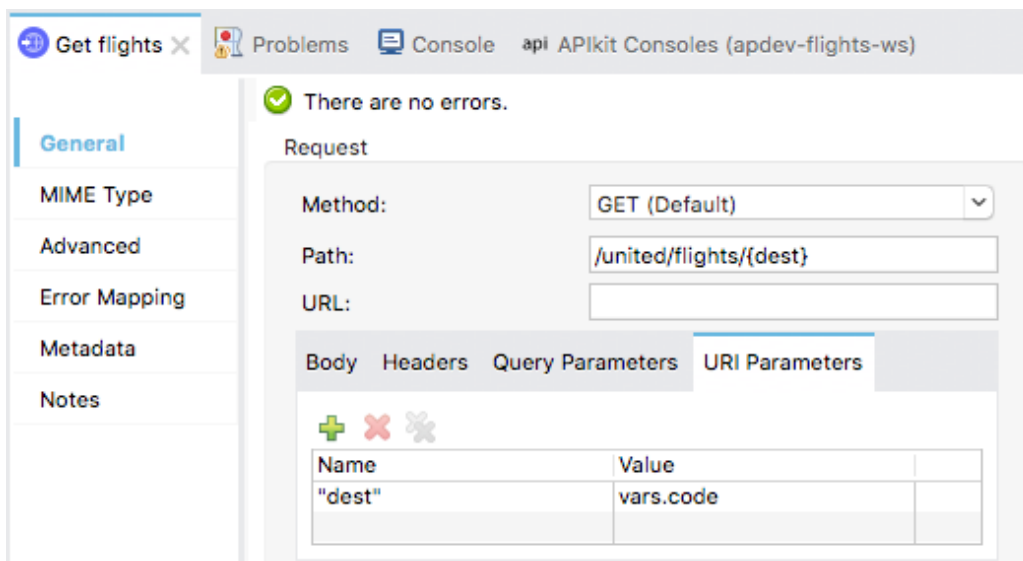


35. Navigate to the Get flights Request properties view in getUnitedFlights.
36. In the Request section, change the path to /united/flights/{dest}.



37. Select the URI Parameters tab.
38. Click the Add button.
39. Set the name to dest.
40. Set the value to the value of the code variable.

`vars.code`



Test the application

41. Save the file to redeploy the application.
42. In Advanced REST Client, make the same request; you should now only get flights to CLE.

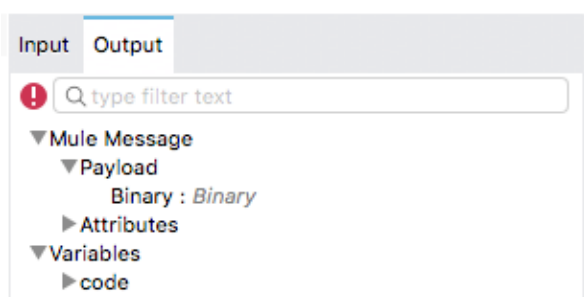


43. Remove the code parameter and make the request; you should now only get results for SFO.

Note: You will transform the data to the format specified by the mua-flights-api in a later walkthrough.

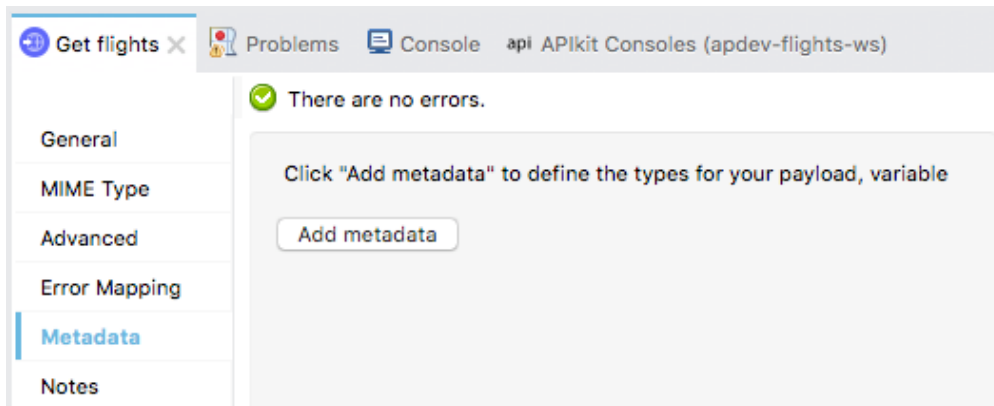
Add metadata for the United Get flights operation response

44. Return to Anypoint Studio.
45. Navigate to the properties view for the Get flights operation in getUnitedFlights.
46. Select the Output tab in the DataSense Explorer and expand Payload; you should see not metadata for the payload.



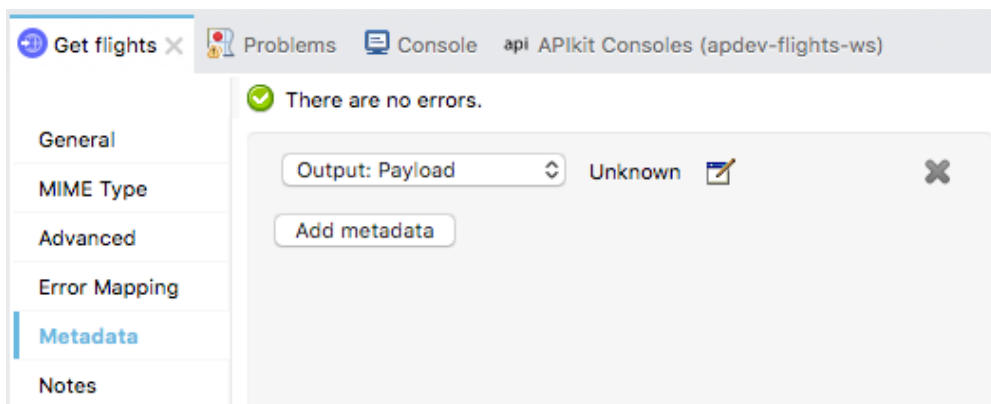
47. In the Get flights properties view, select the Metadata tab.

48. Click the Add metadata button.



49. Change the drop-down menu to Output: Payload.

50. Click the Edit button.



51. In the Select metadata type dialog box, click the Add button.

52. In the Create new type dialog box, set the type id to united_flights_json.

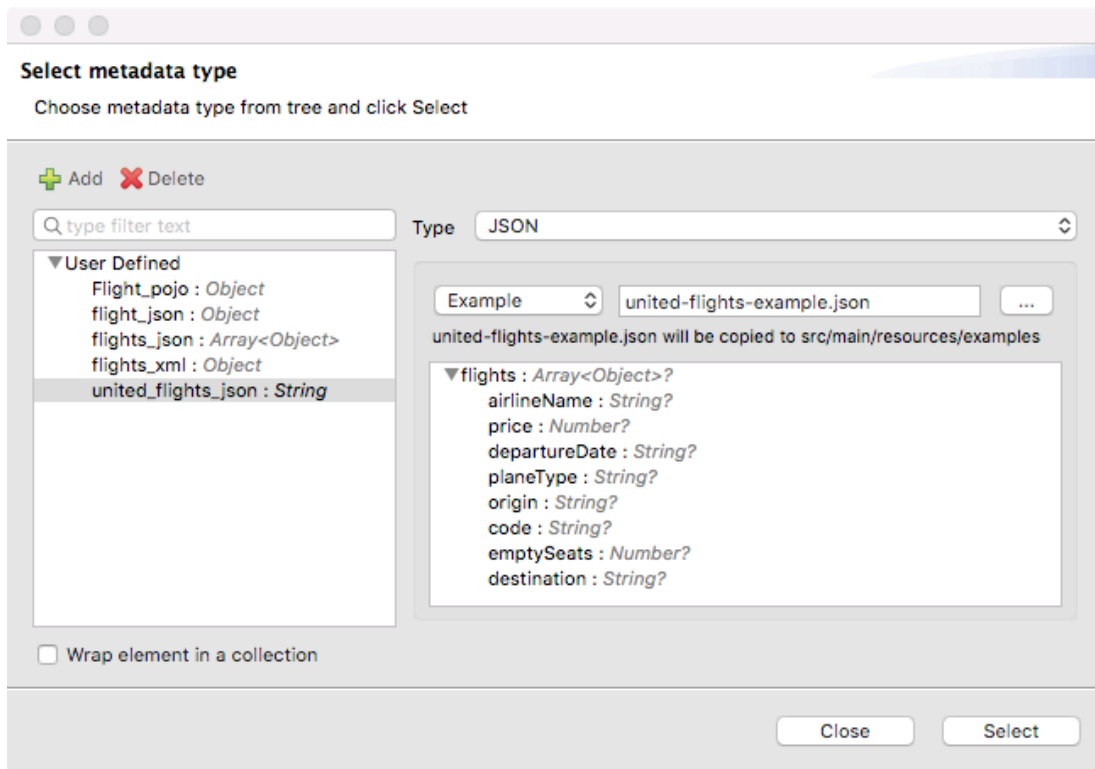
53. Click Create type.

54. In the Select metadata type dialog box, set the type to JSON.

55. Change the Schema selection to Example.

56. Click the browse button and navigate to the projects's src/test/resources folder.

57. Select `united_flights-example.json` and click Open; you should see the example data for the metadata type.



58. Click Select.
59. In the DataSense Explorer, select the Output tab and expand Payload; you should now see the structure for the payload.

