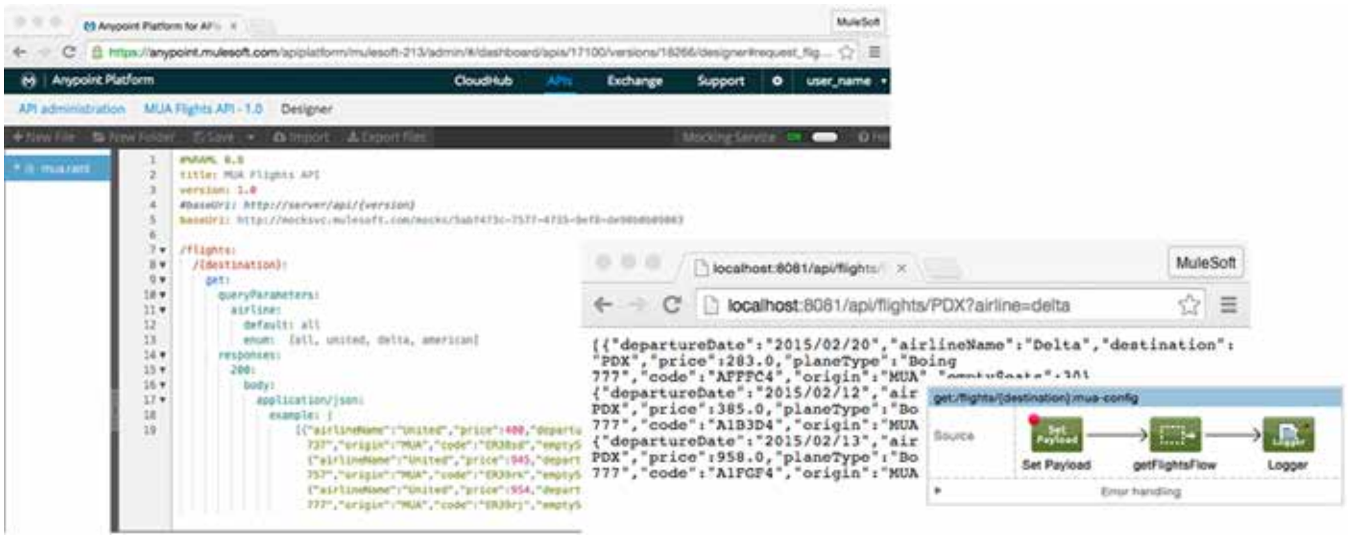
Module 10: Building RESTful Web Services with Anypoint Platform for APIs


Anypoint Platform for APIs
https://anypoint.mulesoft.com/apiplatform/mulesoft-213/admin/#/dashboard/apis/17100/versions/18266/designer#request_fig...
Anypoint Platform
CloudHub
APIs
Exchange
Support
user_name
API administration
MUA Flights API - 1.0
Designer
New File
New Folder
Save
Import
Export files
Mocking Service
mua.raml
localhost:8081/api/flights/PDX?airline=delta
MuleSoft
get:/flights/{destination}:mua-config
Source
Set Payload
getFlightsFlow
Logger
Error handling

```
#%RAML 0.8
title: MUA Flights API
version: 1.0
#baseUri: http://server/api/{version}
baseUri: http://mocksvc.mulesoft.com/mocks/5abf473c-7577-4735-9ef8-de980d0d0983

/flights:
  /{destination}:
    get:
      queryParameters:
        airline:
          default: all
          enum: [all, united, delta, american]
      responses:
        200:
          body:
            application/json:
              example: |
                [{"airlineName":"United","price":400,"departu
                737","origin":"MUA","code":"ER38sd","emptyS
                {"airlineName":"United","price":945,"depart
                757","origin":"MUA","code":"ER39rk","emptyS
                {"airlineName":"United","price":954,"depart
                777","origin":"MUA","code":"ER39rj","emptyS
```

```
[{"departureDate":"2015/02/20","airlineName":"Delta","destination":
"PDX","price":283.0,"planeType":"Boing
777","code":"AFFFC4","origin":"MUA","emptySeats":101
{"departureDate":"2015/02/12","air
PDX","price":385.0,"planeType":"Bo
777","code":"A1B3D4","origin":"MUA
{"departureDate":"2015/02/13","air
PDX","price":958.0,"planeType":"Bo
777","code":"A1FGF4","origin":"MUA
```

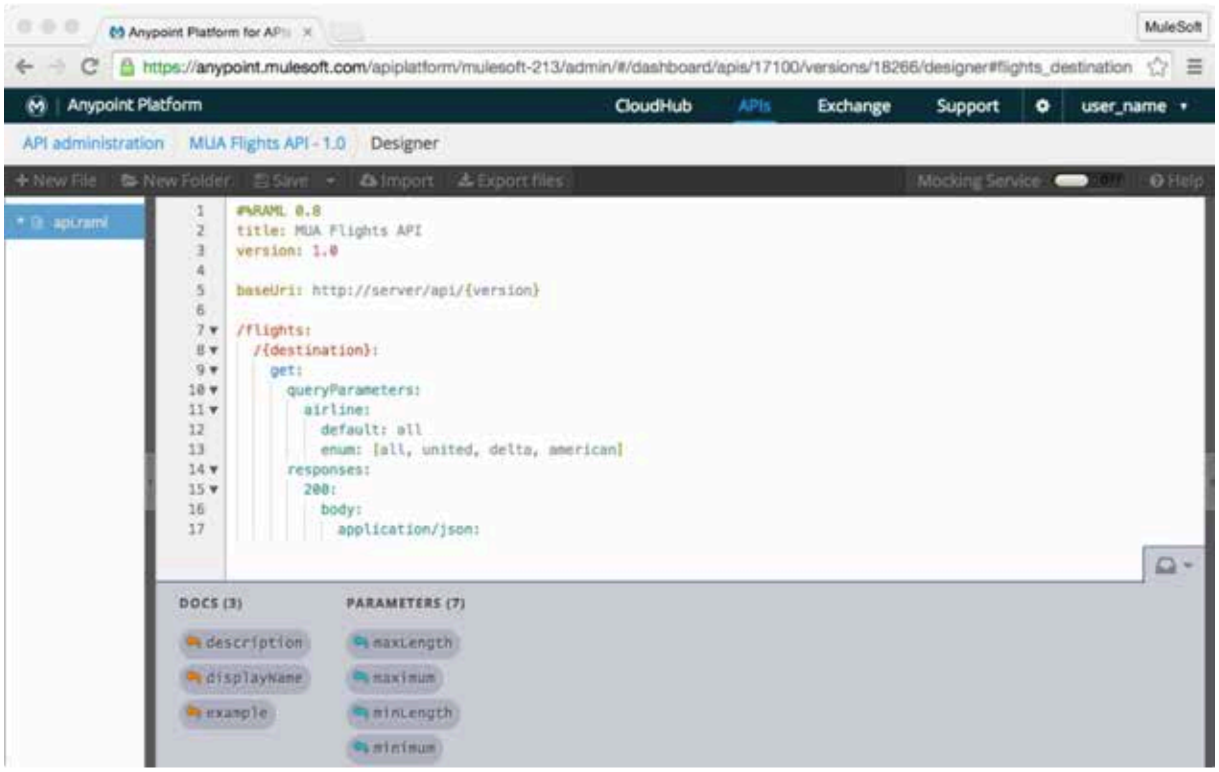
In this module, you will learn:

- To define an API with RAML.
- To create RAML files with Anypoint Designer.
- To implement a RAML file as a RESTful web service with Anypoint Studio and APIkit.

Walkthrough 10-1: Use API Designer to define an API with RAML

In this walkthrough, you will create an API definition for MUA with RAML. You will:

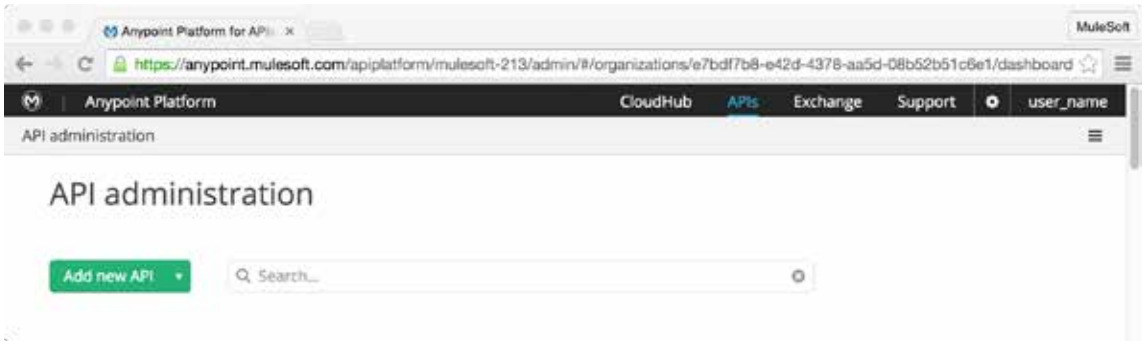
- Add a new API to the Anypoint Platform.
- Use the API Designer to create a RAML file.
- Use a nested resource for the destination and a query parameter for the airline.



```
#%RAML 0.8
title: MUA Flights API
version: 1.0

baseUri: http://server/api/{version}

/flights:
  /{destination}:
    get:
      queryParameters:
        airline:
          default: all
          enum: [all, united, delta, american]
      responses:
        200:
          body:
            application/json:
```

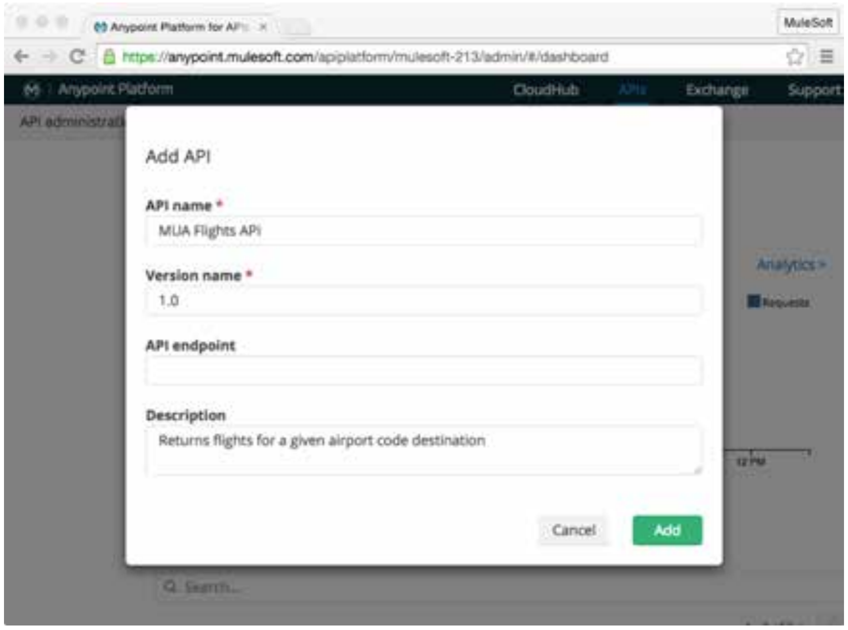
Add a new API

1. In a browser, go to http://anypoint.mulesoft.com and log in.
2. Click the APIs link in the main menu bar.
3. Click the Add new API button.


Anypoint Platform for APIs
MuleSoft
https://anypoint.mulesoft.com/apiplatform/mulesoft-213/admin/#/organizations/e7bdf7b8-e42d-4378-aa5d-08b52b51c6e1/dashboard
Anypoint Platform
CloudHub
APIs
Exchange
Support
user_name
API administration
API administration
Add new API
Search...

4. In the Add API dialog box, enter the following and then click Add.

   - API name: MUA Flights API
   - Version name: 1.0
   - Description: Returns flights for a given airport code destination



Add API

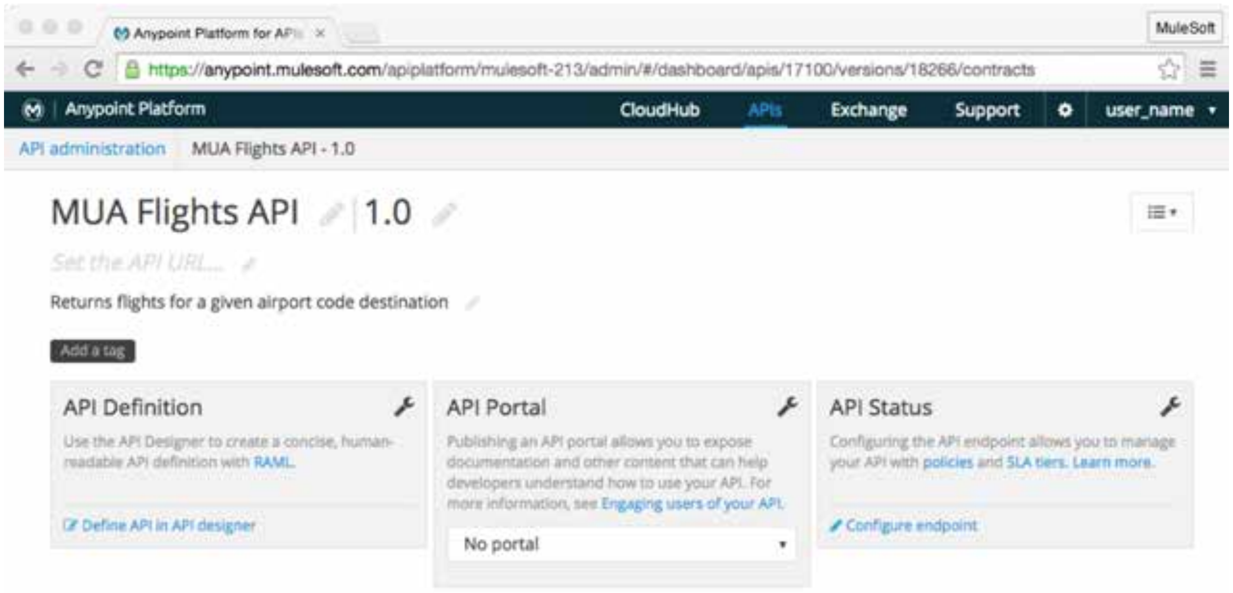
API name *
MUA Flights API

Version name *
1.0

API endpoint

Description
Returns flights for a given airport code destination

5. Take a look at the different sections and links for the API in the API administration.



Anypoint Platform for APIs
MuleSoft
https://anypoint.mulesoft.com/apiplatform/mulesoft-213/admin/#/dashboard/apis/17100/versions/18266/contracts
Anypoint Platform
CloudHub
APIs
Exchange
Support
user_name
API administration
MUA Flights API - 1.0

MUA Flights API 1.0

Set the API URL...

Returns flights for a given airport code destination

API Definition

Use the API Designer to create a concise, human-readable API definition with RAML.

Define API in API designer

API Portal

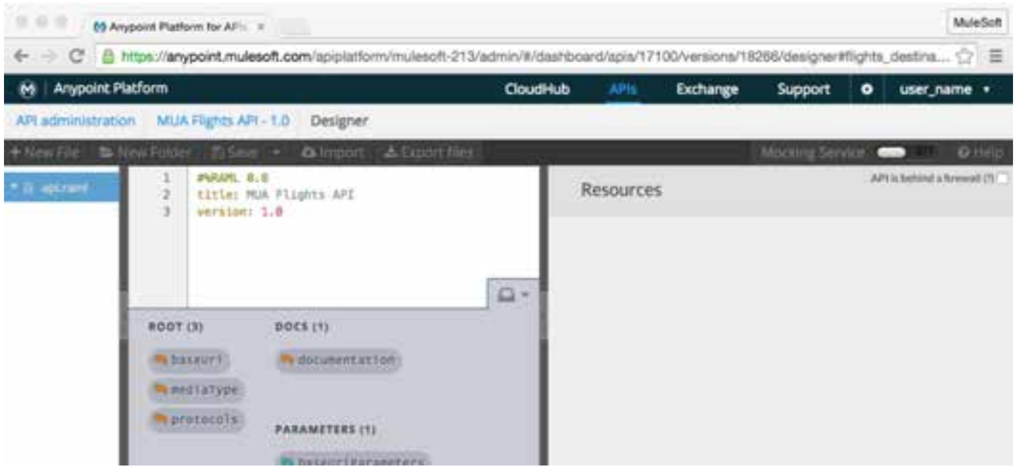
Publishing an API portal allows you to expose documentation and other content that can help developers understand how to use your API. For more information, see Engaging users of your API.

No portal

API Status

Configuring the API endpoint allows you to manage your API with policies and SLA tiers. Learn more.

6. Click the Define API in API designer link; the API Designer should open.



```
#%RAML 0.8
title: MUA Flights API
version: 1.0
```

Note: To change the background color from black to white, press Ctrl+Shift+T.

Add RAML root metatdata

7. Place the cursor on a new line of code at the end of the file.
8. Click the baseUri element in the API Designer shelf to add it to the RAML file.

Note: Leave the default value #baseUri: http://server/api/{version} for now during development.

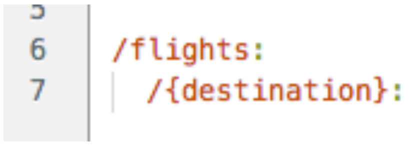
Add a RAML resource

9. Go to a new line of code and add a resource called flights.

```
/flights:
```

Add a nested RAML resource

10. Indent by pressing the Enter key and then the Tab key.
11. Add a nested resource to return flights for a particular destination.

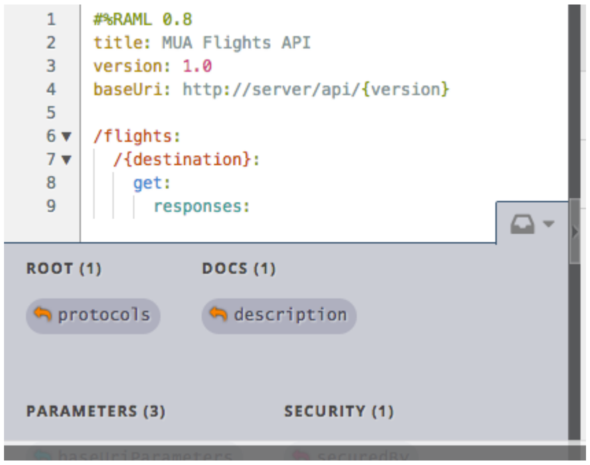


```
/flights:
  /{destination}:
```

Add a RAML method

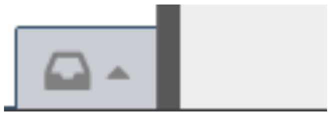
12. Go to a new line of code, press the Tab key, and then press the G key and then the Enter key to add a get method.
13. Indent by pressing the Enter key and then the Tab key.

14. Scroll down in the API Designer shelf and locate and click responses.



```
#%RAML 0.8
title: MUA Flights API
version: 1.0
baseUri: http://server/api/{version}

/flights:
  /{destination}:
    get:
      responses:
```

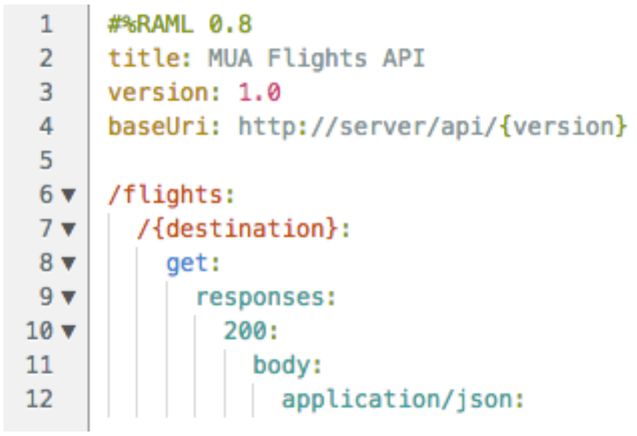
Note: If you don't see the API Designer shelf, it is either minimized or there is an error in your code. To check if it is minimized, scroll down to the bottom of the browser window and look for its icon. If you see it, click it to expand it. If you don't see its icon and you also don't see the API Designer console, you probably have an error in your code, like a missing RAML definition.



15. Indent and type 200:.

```
responses:
        200:
```

16. Indent and type b and then press Enter to add a body parameter (or add it from the shelf).
17. Indent and type a and then press Enter to add application/json (or add it from the shelf).

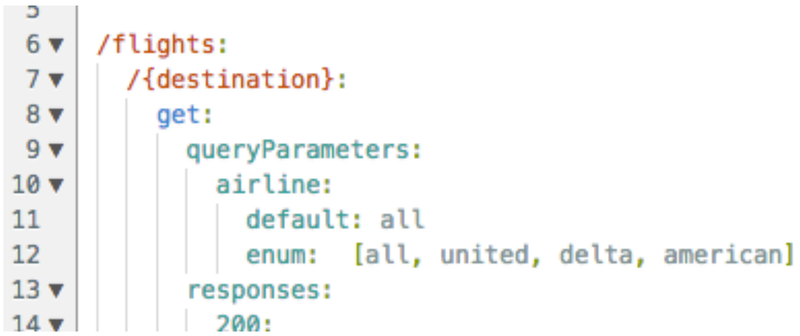


```
#%RAML 0.8
title: MUA Flights API
version: 1.0
baseUri: http://server/api/{version}

/flights:
  /{destination}:
    get:
      responses:
        200:
          body:
            application/json:
```

Note: This is the minimal RAML file needed for Anypoint Studio and APIkit to generate the RESTful interface.

Add a query parameter

18. Go to a new line of code after get:.
19. Press the q key and then the Enter key to add queryParameters.
20. Indent and add a query parameter called airline.
21. Indent and add a default property and set it to all.
22. Return and add an enum property and set it to an array with values all, united, delta, and american.



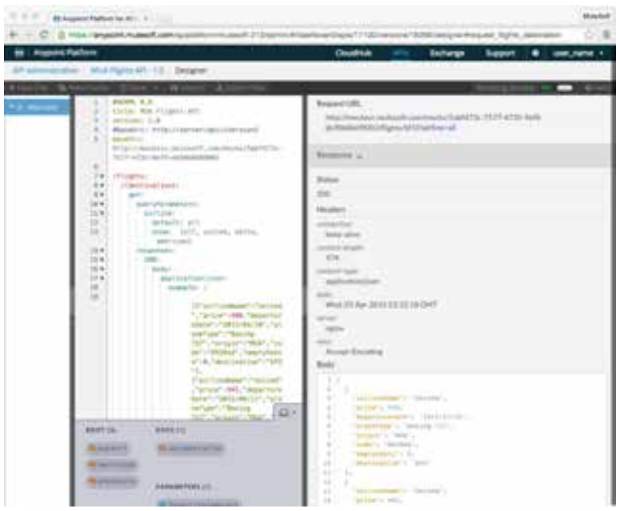
```
/flights:
  /{destination}:
    get:
      queryParameters:
        airline:
          default: all
          enum:  [all, united, delta, american]
      responses:
        200:
```

23. Click the Save button to save the RAML file.

Walkthrough 10-2: Use API Designer to simulate an API

In this walkthrough, you will add example responses to the API definition and test it by running a live simulation. You will:

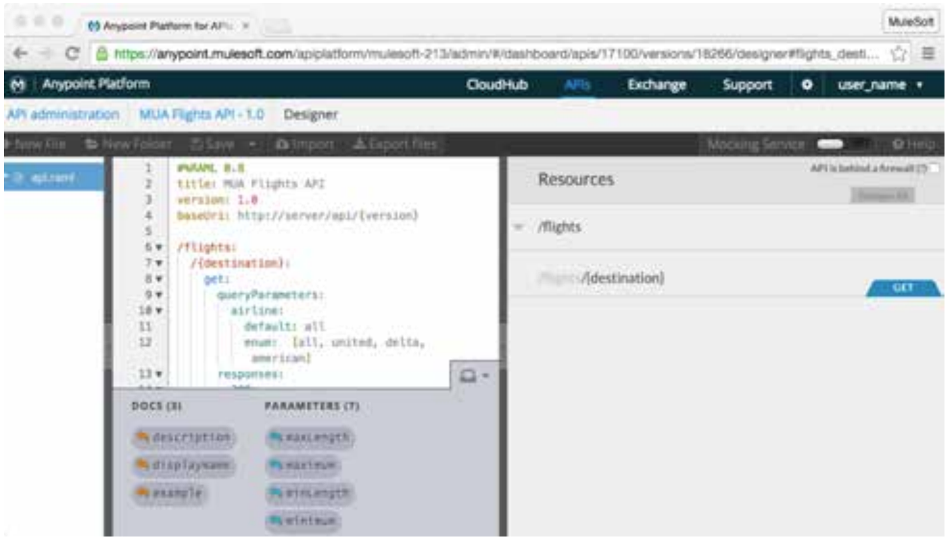
- Use the API console in API Designer.
- Use RAML to specify example responses for your API.
- Use the API Designer mocking service to run a live simulation of your API.



Use the API Designer console

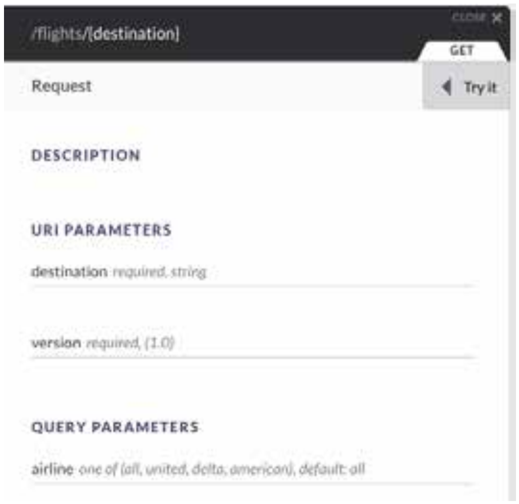
1. Return to your MUA Flights API in API Designer.
2. Look at the API console.

   Note: If you do not see the console, click the arrow located in the middle of the right edge of the browser window.



```
#%RAML 0.8
title: MUA Flights API
version: 1.0
baseUri: http://server/api/{version}

/flights:
  /{destination}:
    get:
      queryParameters:
        airline:
          default: all
          enum:  [all, united, delta,
           american]
      responses:
```

3. Click the GET button for the /flights/{destination} resource; you should see the request information.


/flights/{destination}

Request

DESCRIPTION

URI PARAMETERS

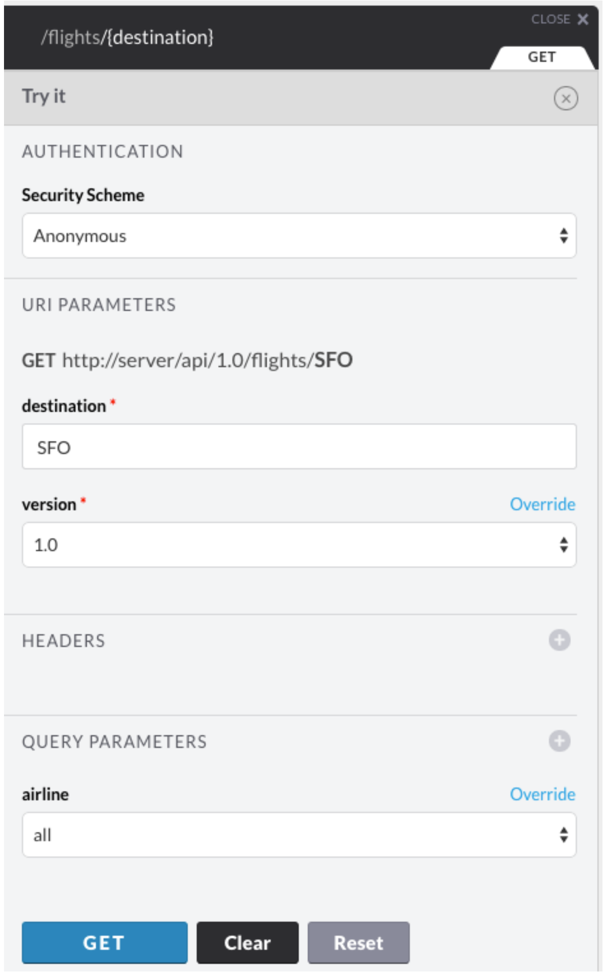
destination *required, string*

version *required, (1.0)*

QUERY PARAMETERS

airline *one of (all, united, delta, american), default: all*

4. Scroll down to the Responses section and see the specification for a 200 status code.
5. Scroll back up and click the Try it button.
6. Enter a destination of SFO and click the GET button.


/flights/{destination}

Try it

AUTHENTICATION

Security Scheme: Anonymous

URI PARAMETERS

GET http://server/api/1.0/flights/**SFO**

destination *: SFO

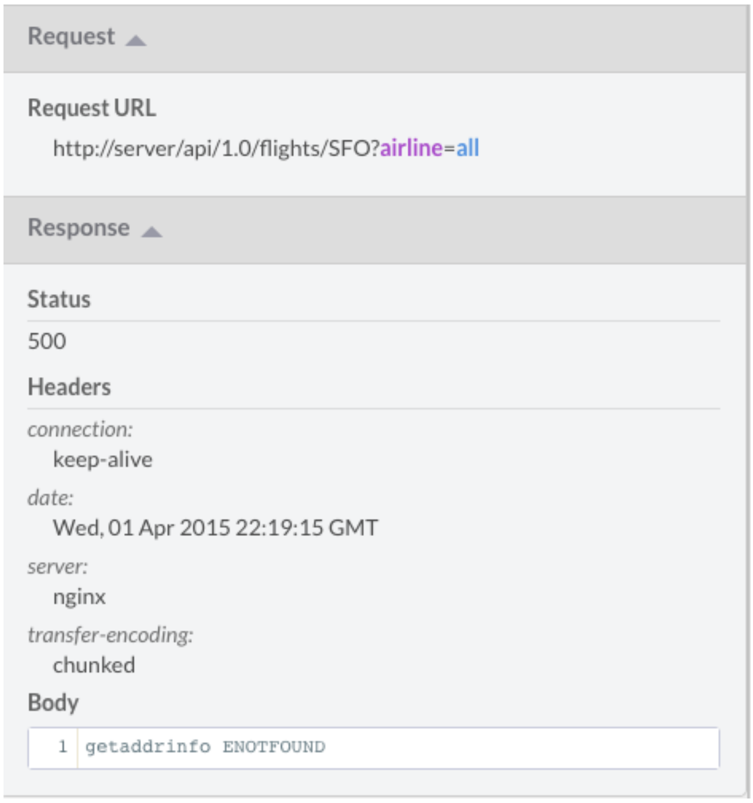
version *: 1.0

HEADERS

QUERY PARAMETERS

airline: all

7. Look at the response; you should get a 500 status code because the baseUri is not found.



```
http://server/api/1.0/flights/SFO?airline=all
```

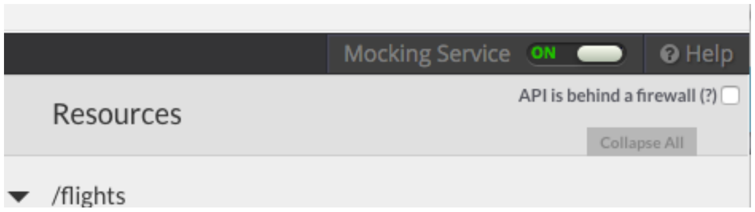
500

```
connection:
  keep-alive
date:
  Wed, 01 Apr 2015 22:19:15 GMT
server:
  nginx
transfer-encoding:
  chunked
```

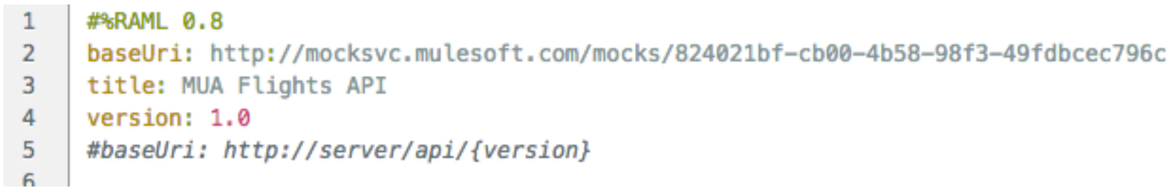
```
getaddrinfo ENOTFOUND
```

Use the mocking service

8. Locate the Mocking Service slider in the menu bar above the console.
9. Slide it to on.



10. Look at the new baseUri in the editor.

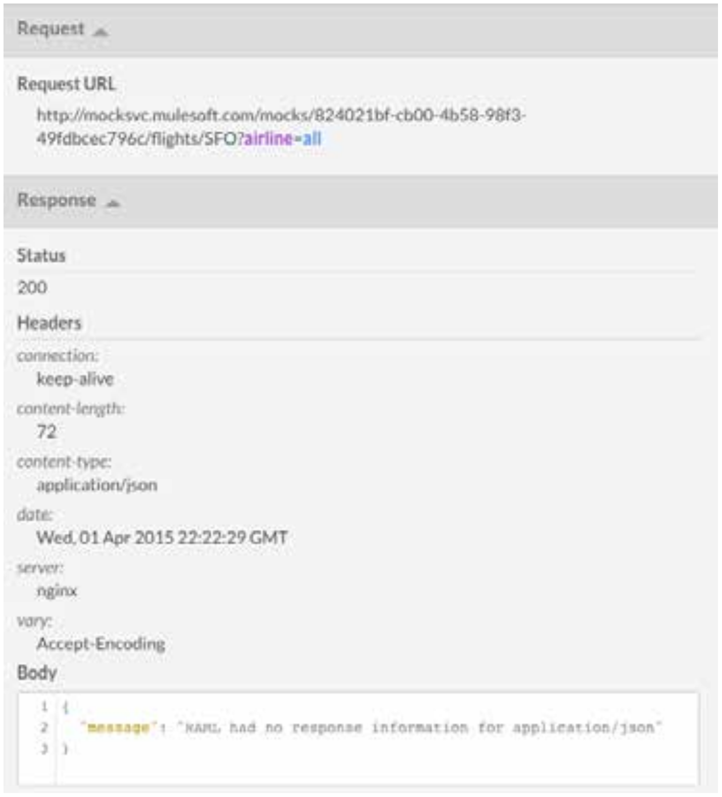


```
#%RAML 0.8
baseUri: http://mocksvc.mulesoft.com/mocks/824021bf-cb00-4b58-98f3-49fdbcec796c
title: MUA Flights API
version: 1.0
#baseUri: http://server/api/{version}
```

11. In the API console, click the Try it button for the /flights/{destination} resource again and then enter a destination and click GET; you should now get a 200 status code.

MuleSoft

12. Scroll down and look at the body; you should get a general RAML message placeholder.



Request

Request URL

http://mocksvc.mulesoft.com/mocks/824021bf-cb00-4b58-98f3-49fdbcec796c/flights/SFO?airline=all

Response

Status

200

Headers

connection:
keep-alive

content-length:
72

content-type:
application/json

date:
Wed, 01 Apr 2015 22:22:29 GMT

server:
nginx

vary:
Accept-Encoding

Body

```
{
  "message": "RAML had no response information for application/json"
}
```

13. Close the /flights/{destination} window.

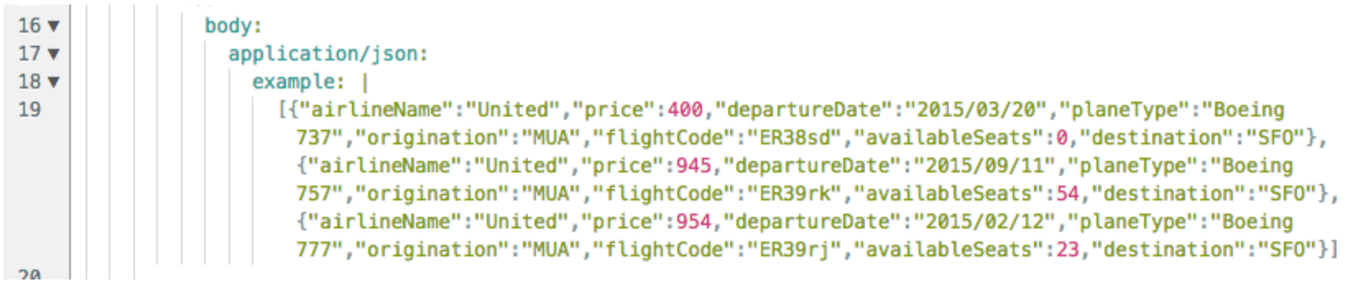
Add examples

14. Return to the course snippets.txt file and copy the Example flights JSON.

```
[{"airlineName":"United","price":400,"departureDate":"2015/03/20",
"planeType":"Boeing 737","origination":"MUA","flightCode":"ER38sd",
"availableSeats":0,"destination":"SFO"},{"airlineName":"United",
"price":945,"departureDate":"2015/09/11","planeType":"Boeing 757",
"origination":"MUA","flightCode":"ER39rk","availableSeats":54,
"destination":"SFO"},{"airlineName":"United","price":954,
"departureDate":"2015/02/12","planeType":"Boeing 777",
"origination":"MUA","flightCode":"ER39rj","availableSeats":23,
"destination":"SFO"}]
```

15. Return to the API Designer editor and indent from the application/json element and add an example element.
16. Add a space and then a | after the example element.

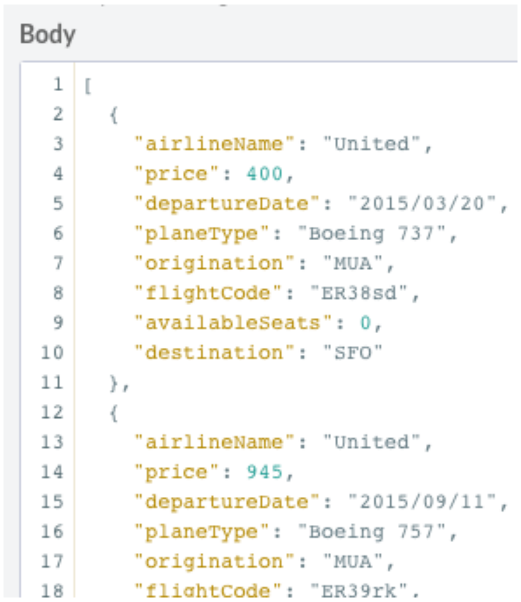
17. Indent and paste the JSON array you just copied.



```
body:
  application/json:
    example: |
      [{"airlineName":"United","price":400,"departureDate":"2015/03/20","planeType":"Boeing
      737","origination":"MUA","flightCode":"ER38sd","availableSeats":0,"destination":"SFO"},
      {"airlineName":"United","price":945,"departureDate":"2015/09/11","planeType":"Boeing
      757","origination":"MUA","flightCode":"ER39rk","availableSeats":54,"destination":"SFO"},
      {"airlineName":"United","price":954,"departureDate":"2015/02/12","planeType":"Boeing
      777","origination":"MUA","flightCode":"ER39rj","availableSeats":23,"destination":"SFO"}]
```

View the example data in the simulation

18. Return to the API console.
19. Click the GET button for the /flights/{destination} resource again.
20. Click Try it, enter a destination, and click GET.
21. Look at the response body; you should now get your example data.

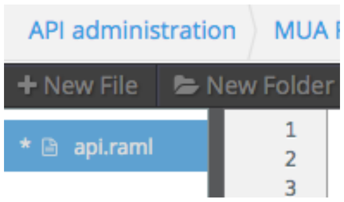


Body

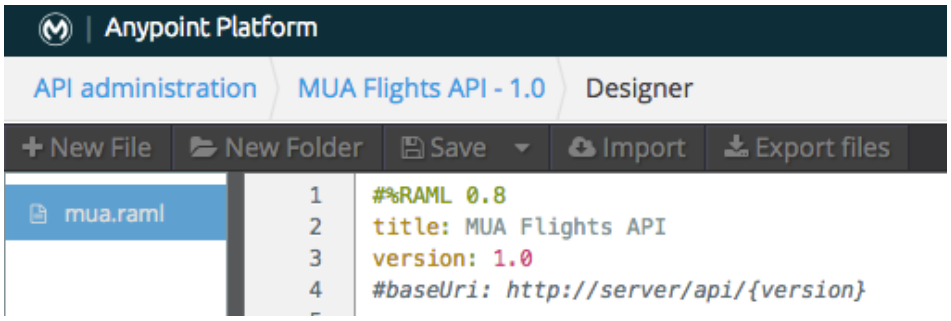
```
[
  {
    "airlineName": "United",
    "price": 400,
    "departureDate": "2015/03/20",
    "planeType": "Boeing 737",
    "origination": "MUA",
    "flightCode": "ER38sd",
    "availableSeats": 0,
    "destination": "SFO"
  },
  {
    "airlineName": "United",
    "price": 945,
    "departureDate": "2015/09/11",
    "planeType": "Boeing 757",
    "origination": "MUA",
    "flightCode": "ER39rk",
```

Save and export the RAML file

22. Slide the Mocking Service slider to off.
23. Save the file.
24. Right-click api.raml in the left pane and select Rename.

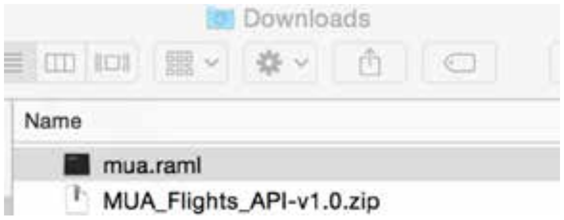


25. Change the name to mua.raml and click OK.
26. Click the Export files button to download the RAML file.



```
#%RAML 0.8
title: MUA Flights API
version: 1.0
#baseUri: http://server/api/{version}
```

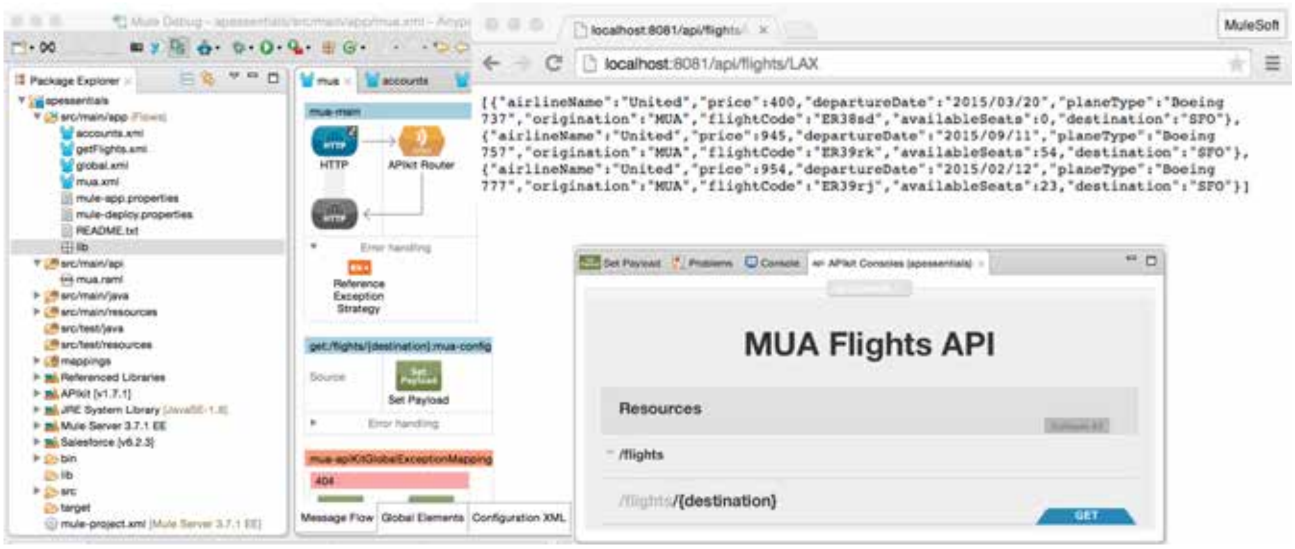
27. Locate the MUA_Flights_API-v1.0.zip file that is downloaded.
28. Expand the ZIP and locate the mua.raml file it contains.



Walkthrough 10-3: Use Anypoint Studio to create a RESTful API interface from a RAML file

In this walkthrough, you will generate a RESTful interface from the RAML file. You will:

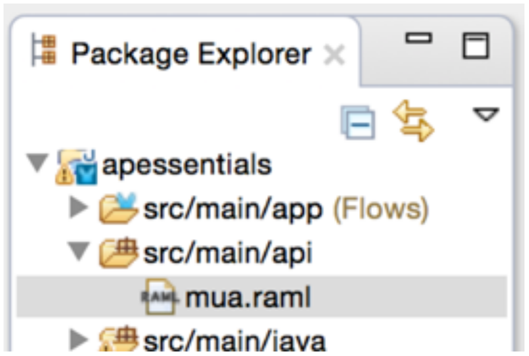
- Add a RAML file to your apessentials project.
- Use Anypoint Studio and APIkit to generate a RESTful web service interface from a RAML file.
- Test the web service in the APIkit Consoles view and a browser.



```
[{"airlineName":"United","price":400,"departureDate":"2015/03/20","planeType":"Boeing
737","origination":"MUA","flightCode":"ER38sd","availableSeats":0,"destination":"SFO"},
{"airlineName":"United","price":945,"departureDate":"2015/09/11","planeType":"Boeing
757","origination":"MUA","flightCode":"ER39rk","availableSeats":54,"destination":"SFO"},
{"airlineName":"United","price":954,"departureDate":"2015/02/12","planeType":"Boeing
777","origination":"MUA","flightCode":"ER39rj","availableSeats":23,"destination":"SFO"}]
```

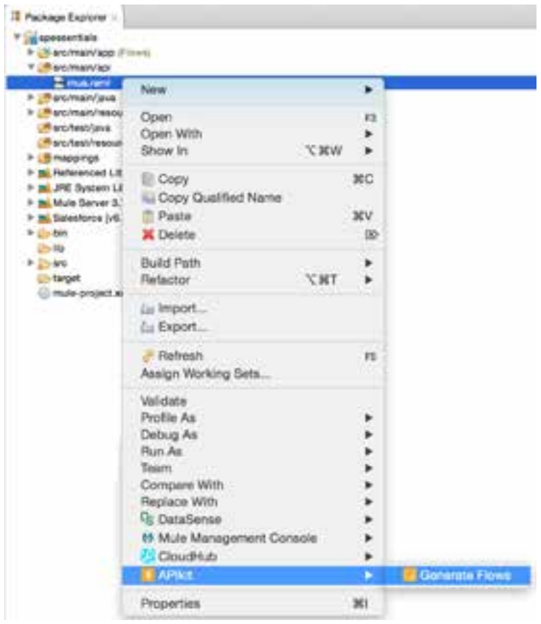
Add the RAML file

1. Drag the mua.raml file into the src/main/api folder in your apessentials project.



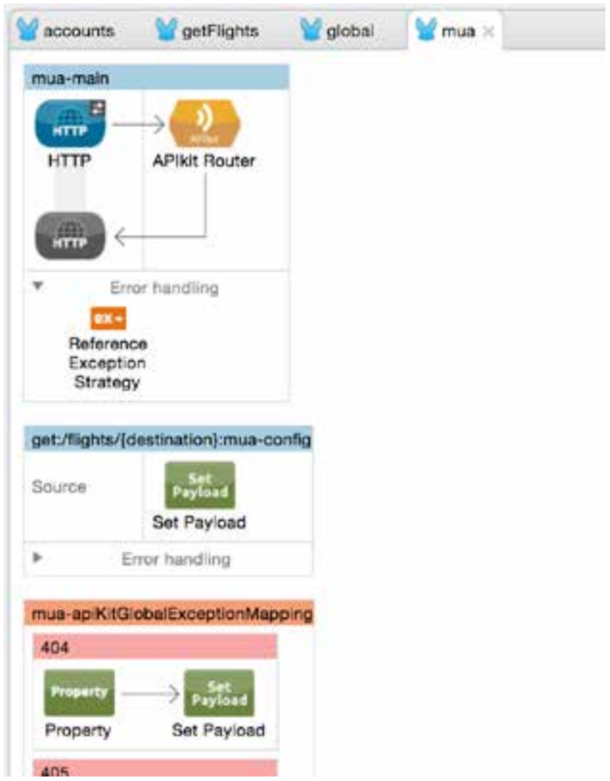
Create a RESTful interface

2. Right-click the RAML file and select APIkit > Generate Flows.



Note: If Generate Flows is disabled, there is an error in your RAML file and you need to return to API Designer and fix it and then re-add it to Anypoint Studio.

3. Wait until a new mua.xml file is generated and opened.

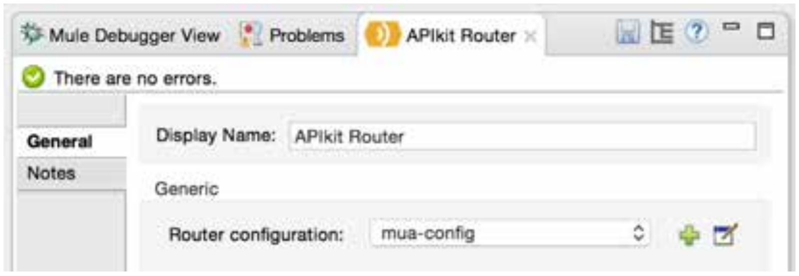


Modify the HTTP endpoint

4. In mua.xml, double-click the HTTP connector in the mua-main flow.
5. Look at the path.
6. Click the Edit button next to Connector Configuration.
7. In the Global Element Properties dialog box, view the host and port values and click OK.
8. Change the connector configuration to the existing HTTP_Listener_Configuration.
9. Switch to the Global Elements view.
10. Select the mua-httpListenerConfig element and click Delete.
11. Return to the Message Flow view.

Examine the APIkit router

12. Double-click the APIkit Router.
13. In the APIkit Router Properties view, see the router configuration is set to mua-config.
14. Click the Edit button next to Connector Configuration.

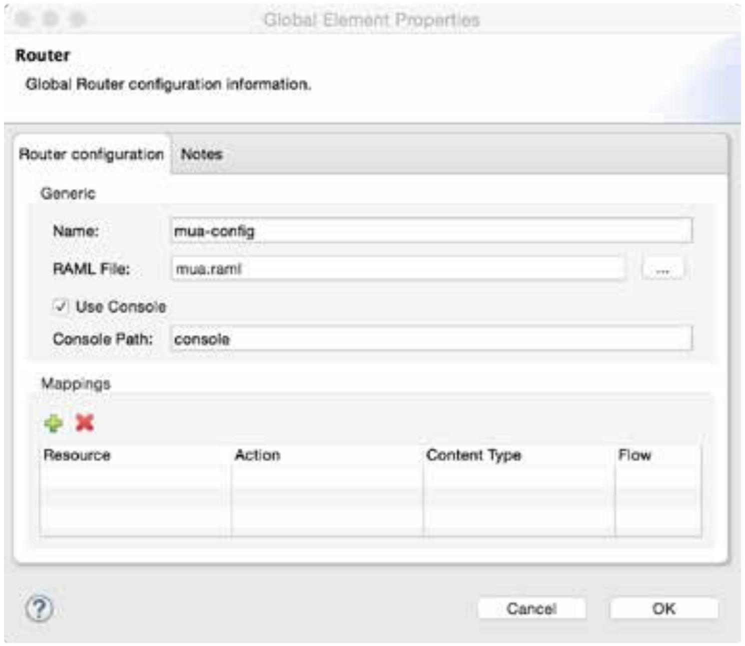


There are no errors.

Display Name: APIkit Router

Router configuration: mua-config

15. In the Global Element Properties dialog box, look at the values and then click OK.



Router
Global Router configuration information.

Name: mua-config

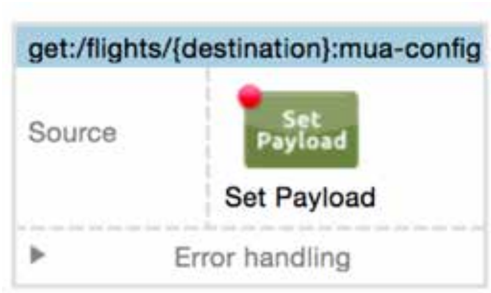
RAML File: mua.raml

☑ Use Console

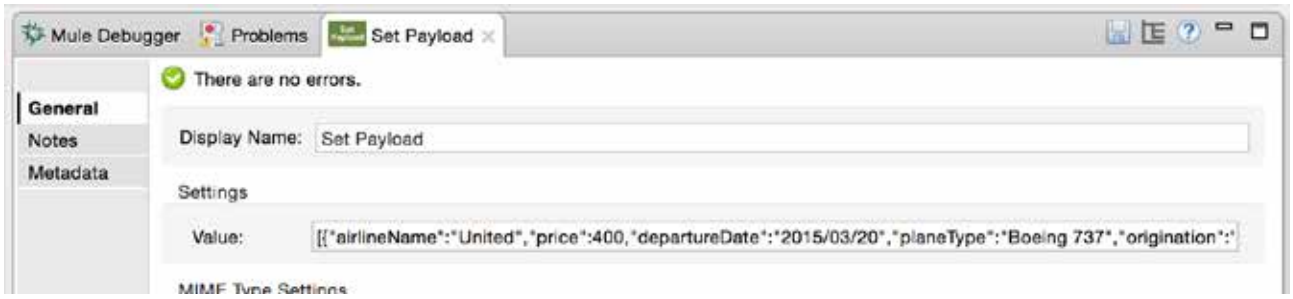
Console Path: console

Look at the generated resource flow

16. Look at the name of the other flow created: get:/flights/{destination}:mua-config.



17. Double-click the Set Payload transformer and look at the value.



There are no errors.

Display Name: Set Payload

Settings

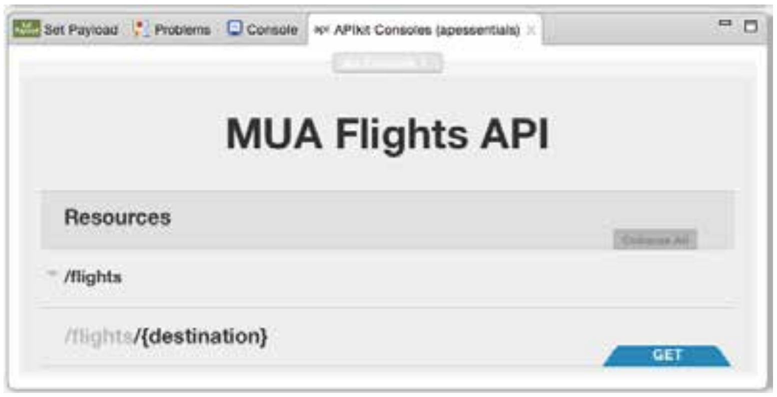
Value: [{"airlineName":"United","price":400,"departureDate":"2015/03/20","planeType":"Boeing 737","origination":"

MIME Type Settings

18. Switch to the Configuration XML view and look at the generated code.

Test the web service in the APIkit Consoles view

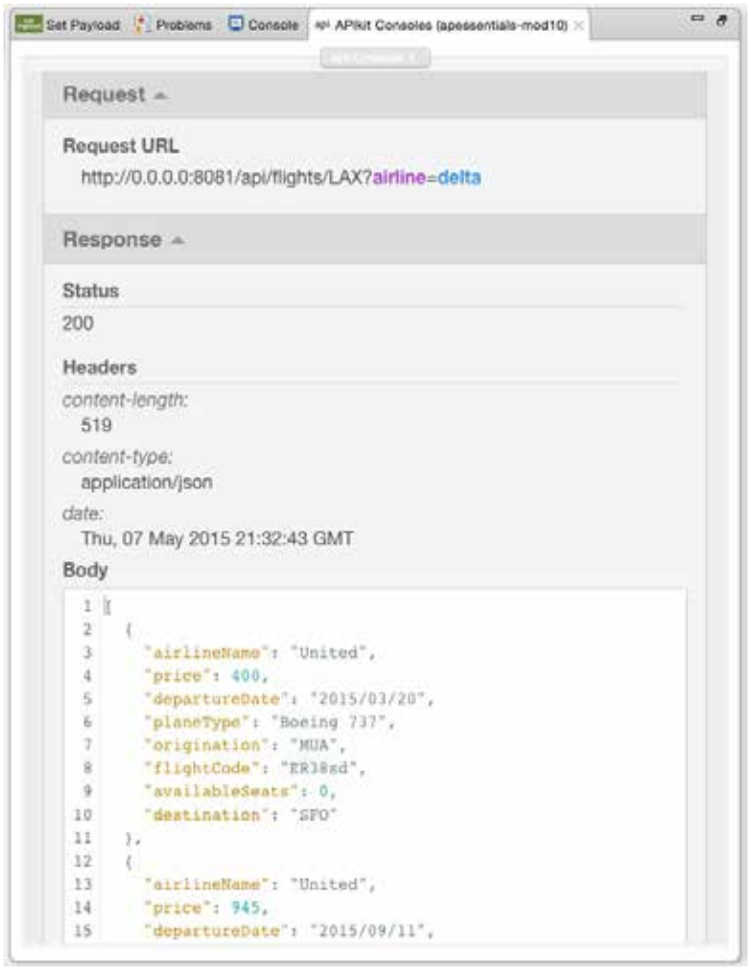
19. Save the file and run the application.
20. Look at the APIkit Consoles view that opens.



Note: If the API is not displayed in the APIkit Console, change your HTTP Listener Configuration host from 0.0.0.0 to localhost.

21. Click the GET button for /flights/{destination}.
22. Click the Try It button.
23. Enter a destination of LAX (or any other value).

24. Click in the airline query parameter field and select one of the enumerated values, like delta.
25. Click the GET button; you should see the example data displayed.



Request

Request URL

http://0.0.0.0:8081/api/flights/LAX?airline=delta

Response

Status

200

Headers

content-length:
519

content-type:
application/json

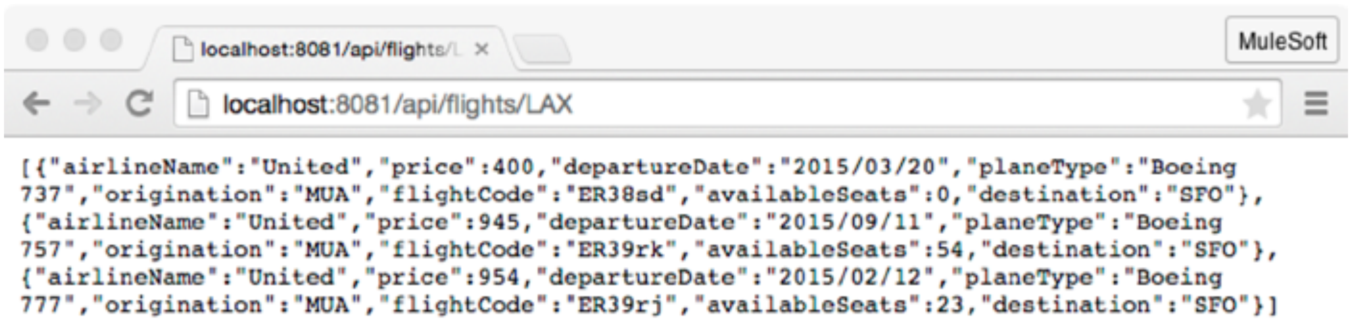
date:
Thu, 07 May 2015 21:32:43 GMT

Body

```
[
  {
    "airlineName": "United",
    "price": 400,
    "departureDate": "2015/03/20",
    "planeType": "Boeing 737",
    "origination": "MUA",
    "flightCode": "ER38sd",
    "availableSeats": 0,
    "destination": "SFO"
  },
  {
    "airlineName": "United",
    "price": 945,
    "departureDate": "2015/09/11",
```

Test the web service in a browser

26. Go to a browser, and make a request to http://localhost:8081/api/flights/LAX (or any other destination); you should see the example data returned.

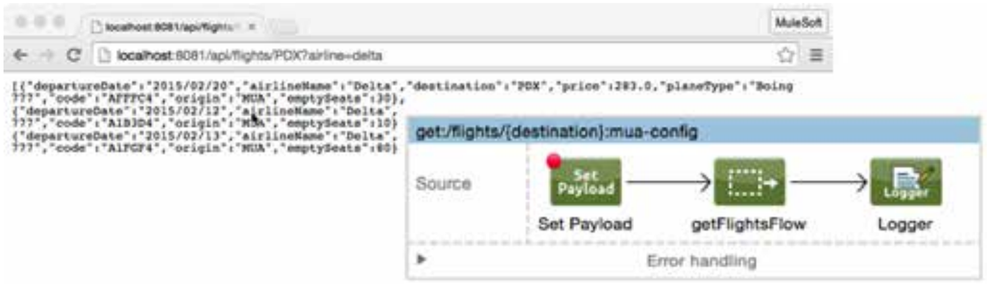


```
[{"airlineName":"United","price":400,"departureDate":"2015/03/20","planeType":"Boeing
737","origination":"MUA","flightCode":"ER38sd","availableSeats":0,"destination":"SFO"},
{"airlineName":"United","price":945,"departureDate":"2015/09/11","planeType":"Boeing
757","origination":"MUA","flightCode":"ER39rk","availableSeats":54,"destination":"SFO"},
{"airlineName":"United","price":954,"departureDate":"2015/02/12","planeType":"Boeing
777","origination":"MUA","flightCode":"ER39rj","availableSeats":23,"destination":"SFO"}]
```

Walkthrough 10-4: Use Anypoint Studio to implement a RESTful web service

In this walkthrough, you will wire the RESTful web service interface up to your back end logic. You will:

- Split the getFlightsFlow into two flows: one that transforms the form post data into a FlightRequest object and the other that gets all the flights.
- Determine how to reference the web service destination and airline values.
- Call the backend flow.
- Test the web service in the APIkit Consoles view and a browser.

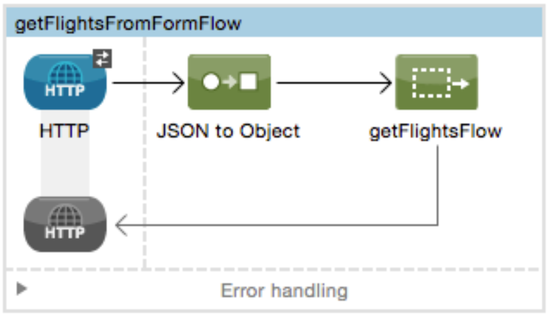


localhost:8081/api/flights/PDX?airline=delta

```
[{"departureDate":"2015/02/20","airlineName":"Delta","destination":"PDX","price":283.0,"planeType":"Boing
777","code":"AFFFC4","origin":"MUA","emptySeats":30},
{"departureDate":"2015/02/12","airlineName":"Delta",
777","code":"A1B3D4","origin":"MUA","emptySeats":10}
{"departureDate":"2015/02/13","airlineName":"Delta",
777","code":"A1FGF4","origin":"MUA","emptySeats":80}
```

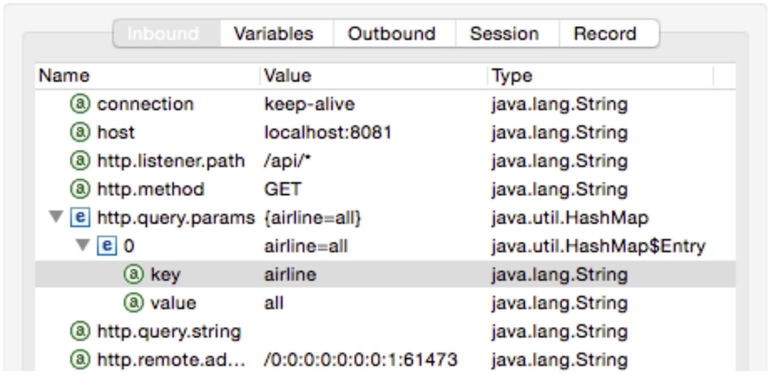
Modify the existing flows

1. Return to getFlights.xml.
2. Drag out a Flow scope element from the palette and drop it at the top of the canvas.
3. Give it a display name of getFlightsFromFormFlow.
4. Drag the HTTP endpoint from getFlightsFlow and drop it in the source section of getFlightsFromFormFlow.
5. Drag the JSON to Object transformer from getFlightsFlow and drop it in the process section of getFlightsFromFormFlow.
6. Drag out a Flow Reference component from the palette and drop it in the process section of getFlightsFromFormFlow.
7. In the Flow Reference Properties view, set the flow name to getFlightsFlow.



Determine how to reference the web service destination and airline values

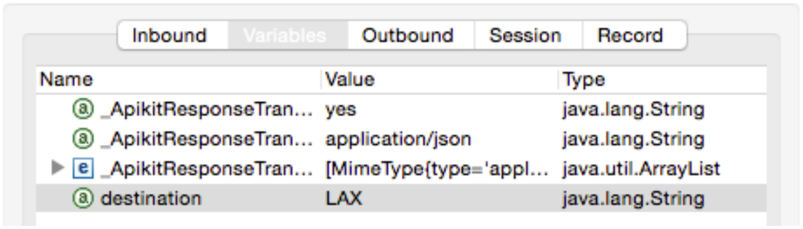
8. Return to mua.xml.
9. Add a Logger to the get:/flights/{destination} flow.
10. Add a breakpoint to the Set Payload transformer in the get:/flights/{destination} flow.
11. Save all the files and debug the application.
12. Make a request to http://localhost:8081/api/flights/LAX.
13. In the Mule Debugger view, look at the inbound properties; you should see a query parameter called airline with a value of all.



Name	Value	Type
connection	keep-alive	java.lang.String
host	localhost:8081	java.lang.String
http.listener.path	/api/*	java.lang.String
http.method	GET	java.lang.String
http.query.params	{airline=all}	java.util.HashMap
0	airline=all	java.util.HashMap$Entry
key	airline	java.lang.String
value	all	java.lang.String
http.query.string		java.lang.String
http.remote.ad...	/0:0:0:0:0:0:0:1:61473	java.lang.String

Note: You may need to remove the breakpoints from the

14. Click the Variables tab; you should see a variable called destination with a value of LAX.

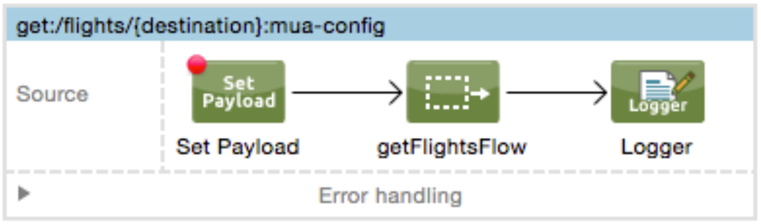


Name	Value	Type
_ApikitResponseTran...	yes	java.lang.String
_ApikitResponseTran...	application/json	java.lang.String
_ApikitResponseTran...	[MimeType{type='appl...	java.util.ArrayList
destination	LAX	java.lang.String

15. Click the Resume button.

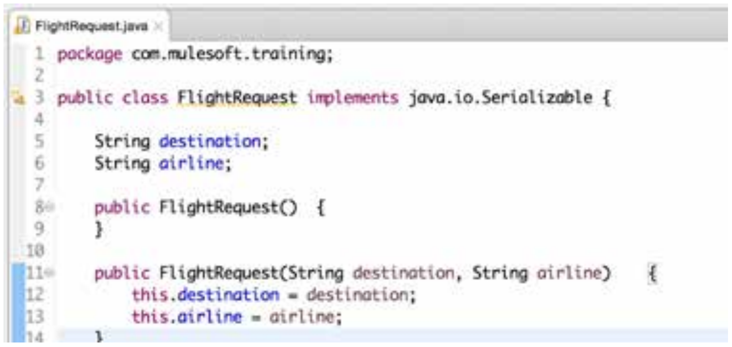
Call the backend flow

16. Return to the get:/flights/{destination} flow.
17. Add a Flow Reference component after the Set Payload transformer.
18. In the Flow Reference Properties view, set the flow name to getFlightsFlow.



Review the FlightRequest Java class

19. Open the com.mulesoft.training.FlightRequest.java class located in the project's src/main/java folder and examine the code.



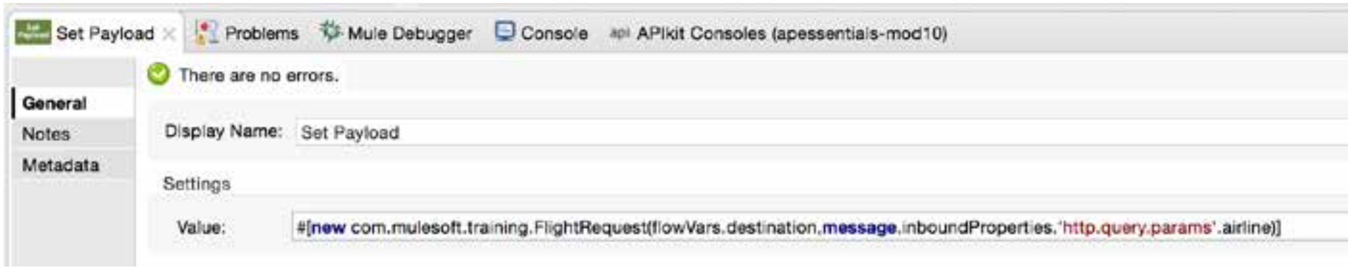
FlightRequest.java

```
package com.mulesoft.training;

public class FlightRequest implements java.io.Serializable {

    String destination;
    String airline;

    public FlightRequest()  {
    }

    public FlightRequest(String destination, String airline)    {
        this.destination = destination;
        this.airline = airline;
    }
```

Set the payload to a Java object with flight request data

20. Double-click the Set Payload transformer in the get:/flights/{destination} flow.
21. In the Properties view, set the value to a new instance of com.mulesoft.training.FlightRequest, passing destination and message.inboundProperties.'http.query.params'.airline to it as arguments.

    Note: You can also copy this expression form the course snippets.txt file.



There are no errors.

Display Name: Set Payload

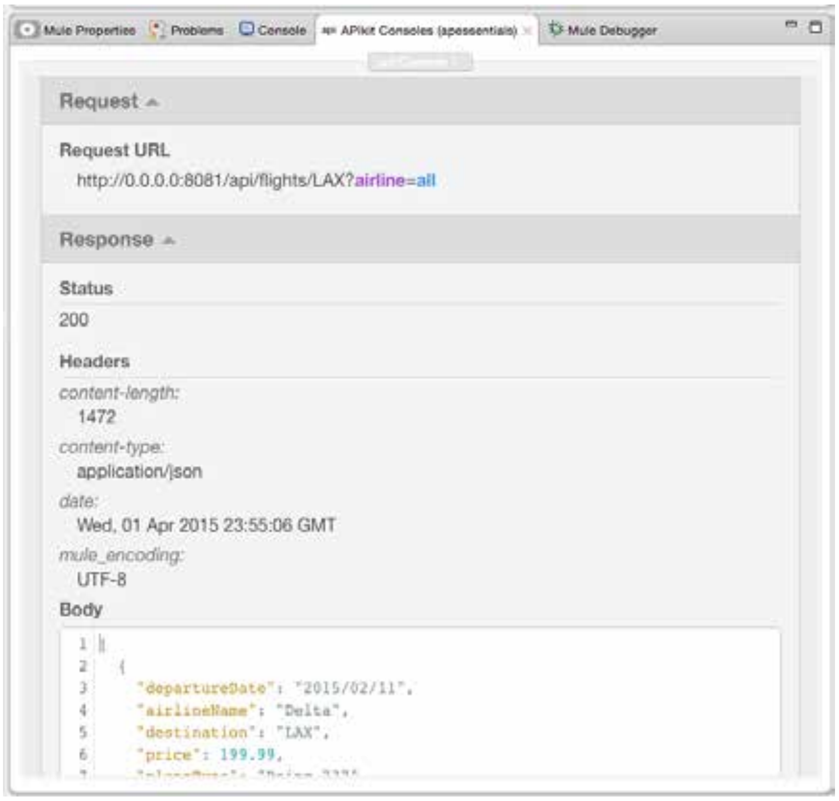
Settings

Value: `#[new com.mulesoft.training.FlightRequest(flowVars.destination,message.inboundProperties.'http.query.params'.airline)]`

Test the web service in the APIkit Consoles view

22. Save the files and run the application.
23. In the APIkit Consoles view, click the GET button for /flights/{destination}.
24. Click the Try It button.

25. Enter a destination of LAX (or any other value) and click the GET button; you should now see the real data (for LAX) displayed.



Request

Request URL
http://0.0.0.0:8081/api/flights/LAX?airline=all

Response

Status
200

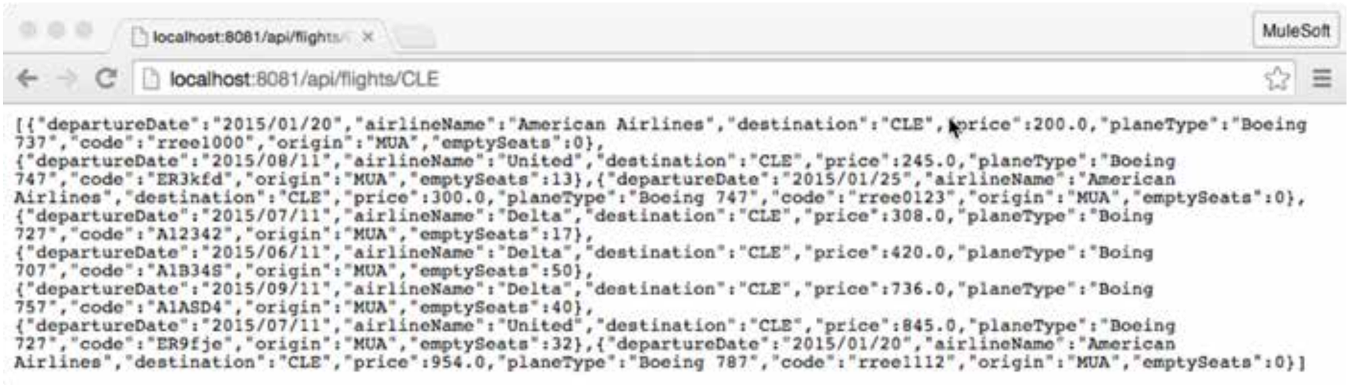
Headers
content-length:
1472
content-type:
application/json
date:
Wed, 01 Apr 2015 23:55:06 GMT
mule_encoding:
UTF-8

Body
```
[
  {
    "departureDate": "2015/02/11",
    "airlineName": "Delta",
    "destination": "LAX",
    "price": 199.99,
```

26. Scroll up in the /flights/{destination} resource window.
27. Set the airline query parameter to united.
28. Click the GET button; you see should now see only the flights for LAX on United.

Test the web service in a browser

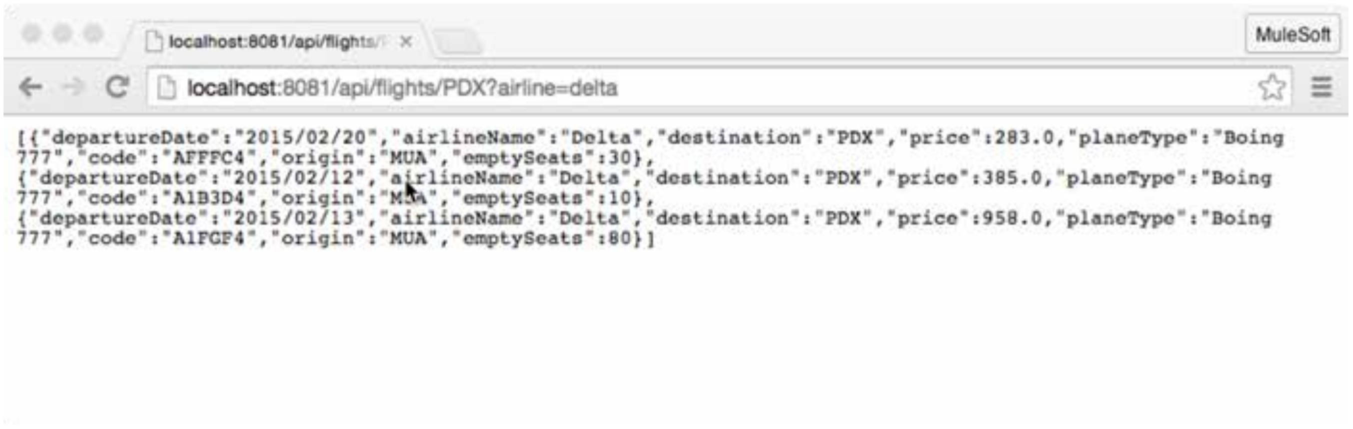
29. Make a request to http://localhost:8081/api/flights/CLE; you should see data for all the flights to CLE returned.



localhost:8081/api/flights/CLE

```
[{"departureDate":"2015/01/20","airlineName":"American Airlines","destination":"CLE","price":200.0,"planeType":"Boeing
737","code":"rree1000","origin":"MUA","emptySeats":0},
{"departureDate":"2015/08/11","airlineName":"United","destination":"CLE","price":245.0,"planeType":"Boeing
747","code":"ER3kfd","origin":"MUA","emptySeats":13},{"departureDate":"2015/01/25","airlineName":"American
Airlines","destination":"CLE","price":300.0,"planeType":"Boeing 747","code":"rree0123","origin":"MUA","emptySeats":0},
{"departureDate":"2015/07/11","airlineName":"Delta","destination":"CLE","price":308.0,"planeType":"Boing
727","code":"A12342","origin":"MUA","emptySeats":17},
{"departureDate":"2015/06/11","airlineName":"Delta","destination":"CLE","price":420.0,"planeType":"Boing
707","code":"A1B34S","origin":"MUA","emptySeats":50},
{"departureDate":"2015/09/11","airlineName":"Delta","destination":"CLE","price":736.0,"planeType":"Boing
757","code":"A1ASD4","origin":"MUA","emptySeats":40},
{"departureDate":"2015/07/11","airlineName":"United","destination":"CLE","price":845.0,"planeType":"Boeing
727","code":"ER9fje","origin":"MUA","emptySeats":32},{"departureDate":"2015/01/20","airlineName":"American
Airlines","destination":"CLE","price":954.0,"planeType":"Boeing 787","code":"rree1112","origin":"MUA","emptySeats":0}]
```

30. Make another request to http://localhost:8081/api/flights/PDX?airline=delta to retrieve flights for PDX on Delta; ensure the correct data is returned.



localhost:8081/api/flights/
MuleSoft
localhost:8081/api/flights/PDX?airline=delta

```
[{"departureDate":"2015/02/20","airlineName":"Delta","destination":"PDX","price":283.0,"planeType":"Boing
777","code":"AFFFC4","origin":"MUA","emptySeats":30},
{"departureDate":"2015/02/12","airlineName":"Delta","destination":"PDX","price":385.0,"planeType":"Boing
777","code":"A1B3D4","origin":"MUA","emptySeats":10},
{"departureDate":"2015/02/13","airlineName":"Delta","destination":"PDX","price":958.0,"planeType":"Boing
777","code":"A1FGF4","origin":"MUA","emptySeats":80}]
```