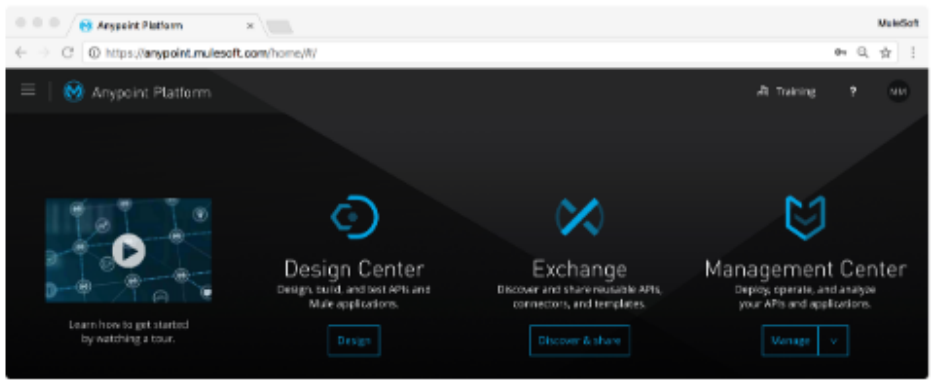

MuleSoft
Module 2: Introducing Anypoint Platform

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

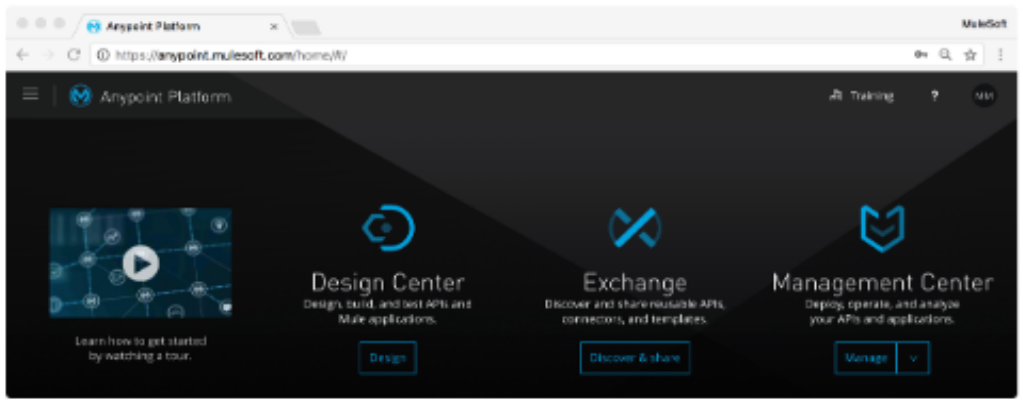
- Describe the benefits of Anypoint Platform and MuleSoft's approach to be successful with it
- Describe the role of each component in building application networks
- Navigate Anypoint Platform
- Locate APIs and other assets needed to build integrations and APIs in Anypoint Exchange
- Build basic integrations to connect systems using flow designer



All contents © MuleSoft Inc.

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

- Describe the benefits of Anypoint Platform and MuleSoft's approach to be successful with it
- Describe the role of each component in building application networks
- Navigate Anypoint Platform
- Locate APIs and other assets needed to build integrations and APIs in Anypoint Exchange
- Build basic integrations to connect systems using flow designer



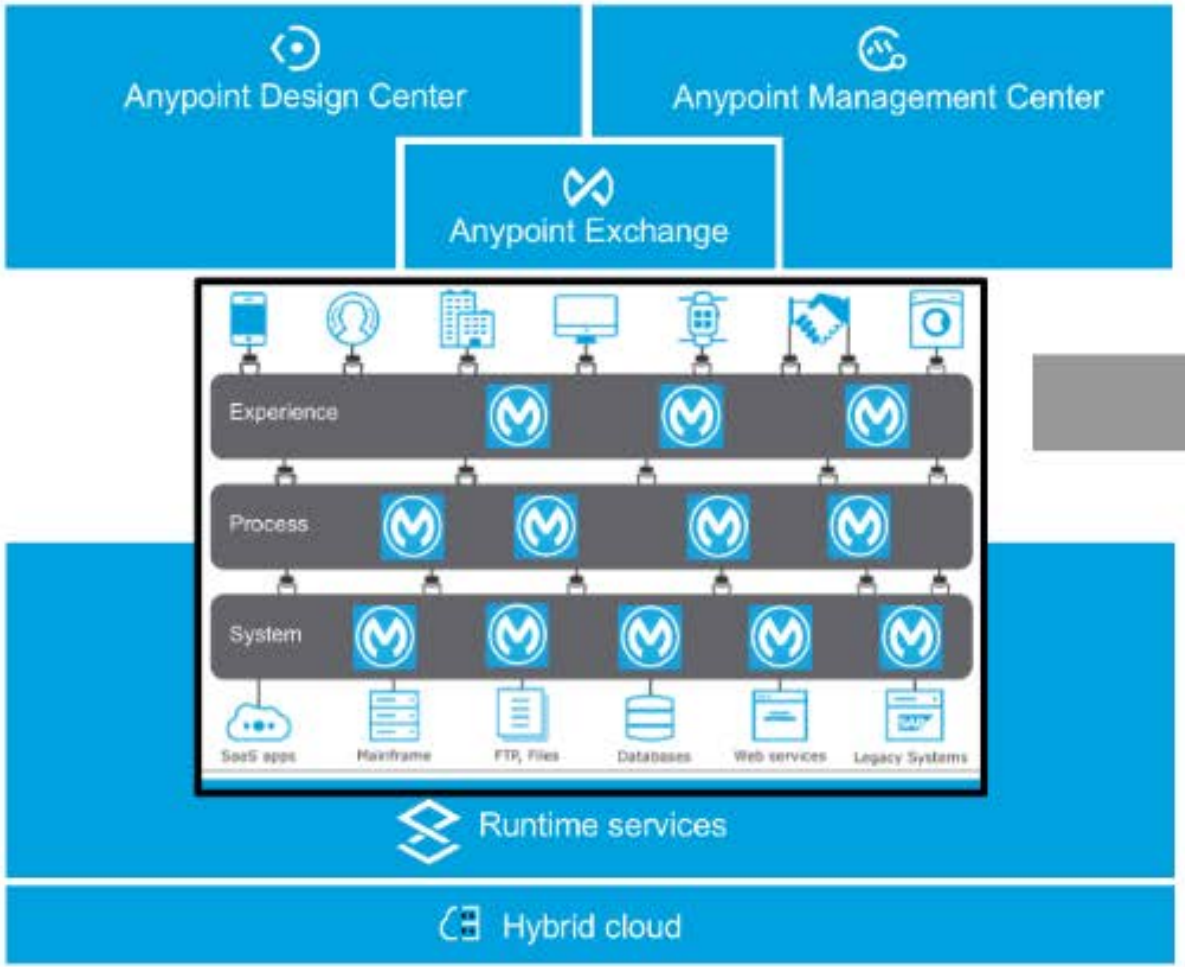
All contents © MuleSoft Inc.

Introducing Anypoint Platform

Anypoint Platform uniquely enables the building of an application network

MuleSoft

Anypoint Platform



Application network

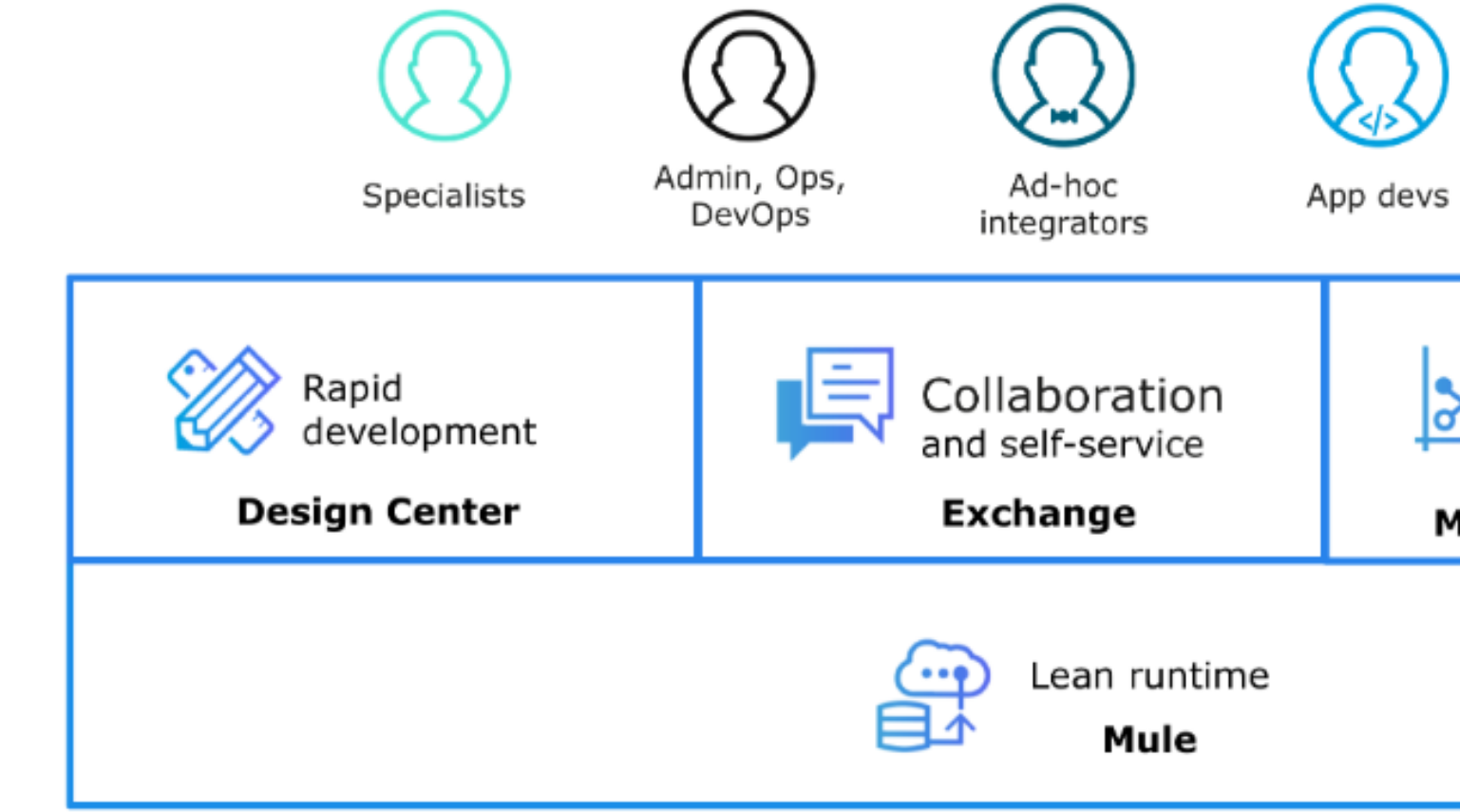
All contents © MuleSoft Inc.

The most advanced enterprise platform for designing, developing and managing APIs and integrations

- Uniquely built as a single product
- Deploy anywhere
- Wide range of use cases

All contents © MuleSoft Inc.

1 Design Center, 1 Mgmt Center, 1 Runtime



All contents © MuleSoft Inc.

Design once, deploy anywhere

Anypoint Design Center

Anypoint Management Center

On-prem

Private cloud

fully managed iPaaS

Databases

FTP, Files

Web services

SaaS Apps

On-prem Apps

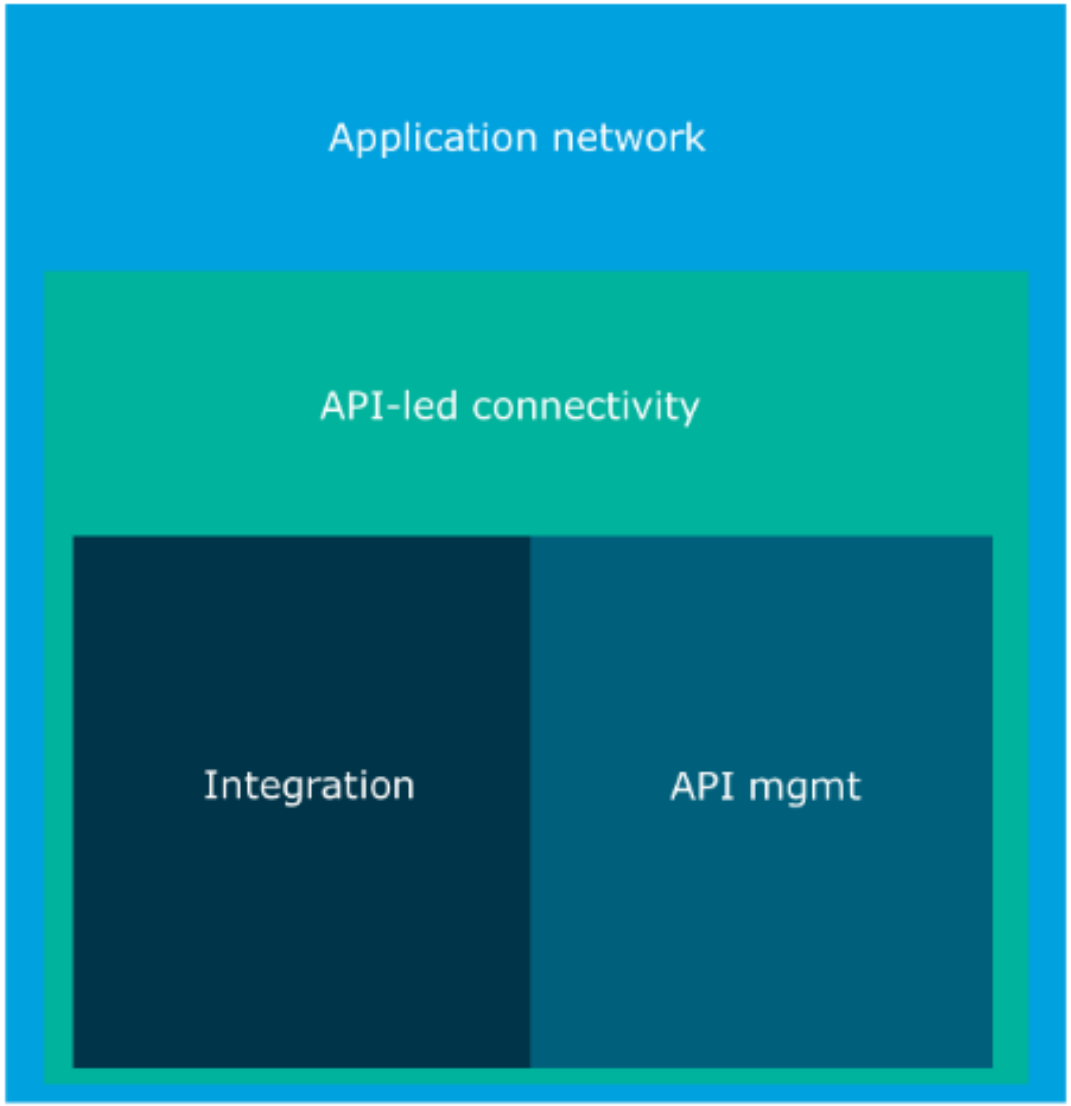
Social Apps

IoT

Partners

All contents © MuleSoft Inc.

Unique benefits for each layer

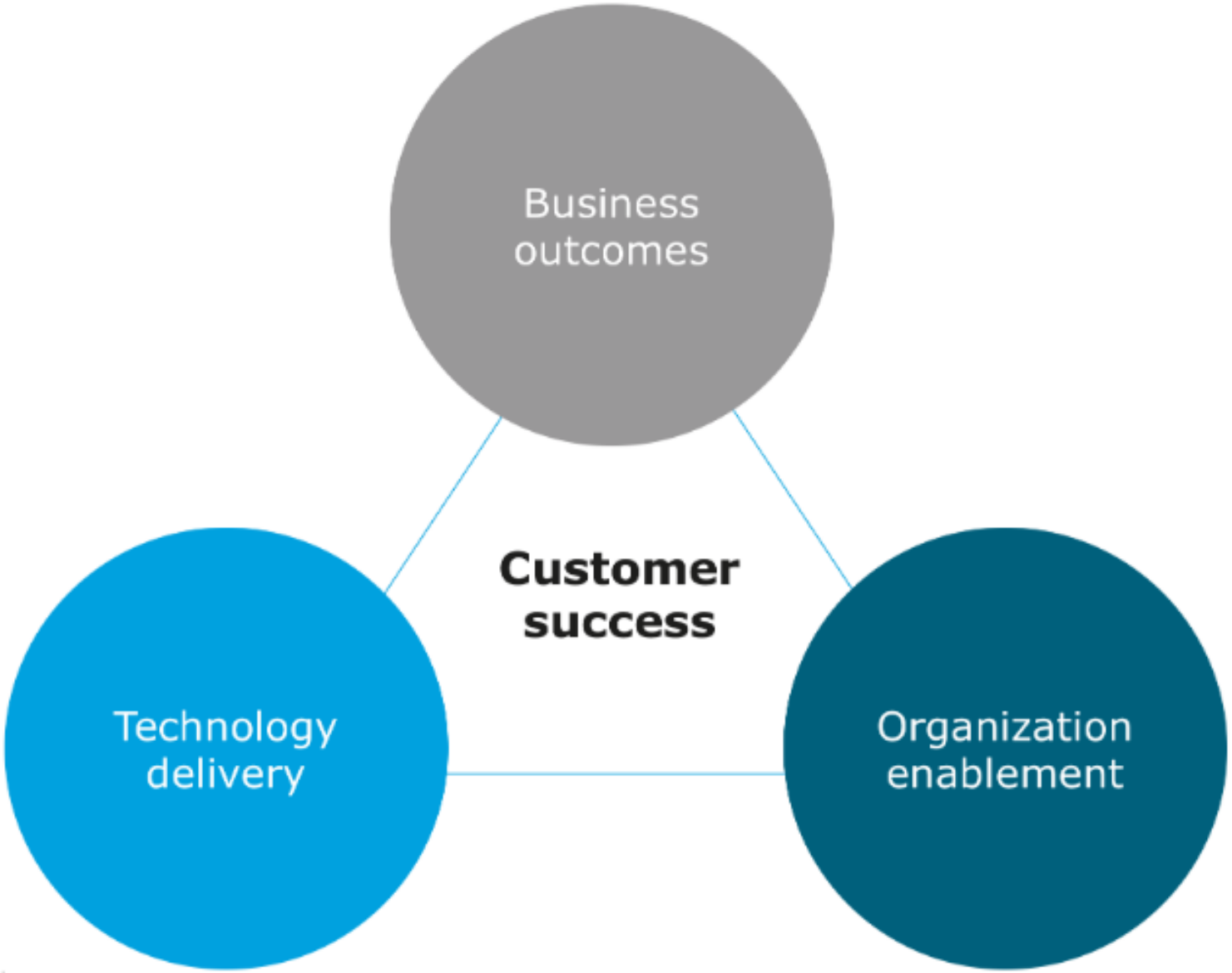


- Speed of delivery
- Actionable visibility
- Secure by design
- Future-proof architecture
- Intentional self-service

All contents © MuleSoft Inc.

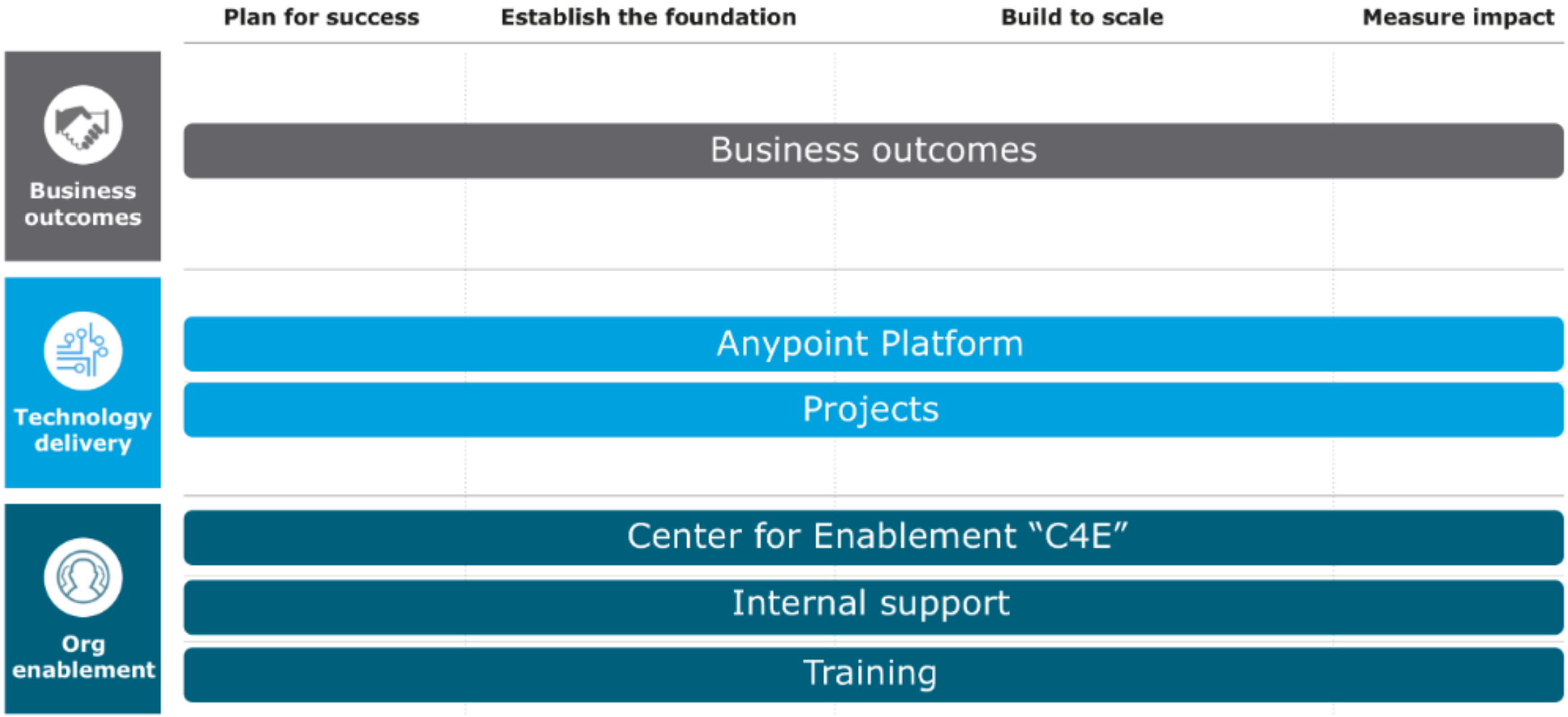
Achieving success with Anypoint Platform

MuleSoft's approach is centered around 3 core pillars



All contents © MuleSoft Inc.

Defining the path to achieve success



All contents © MuleSoft Inc.

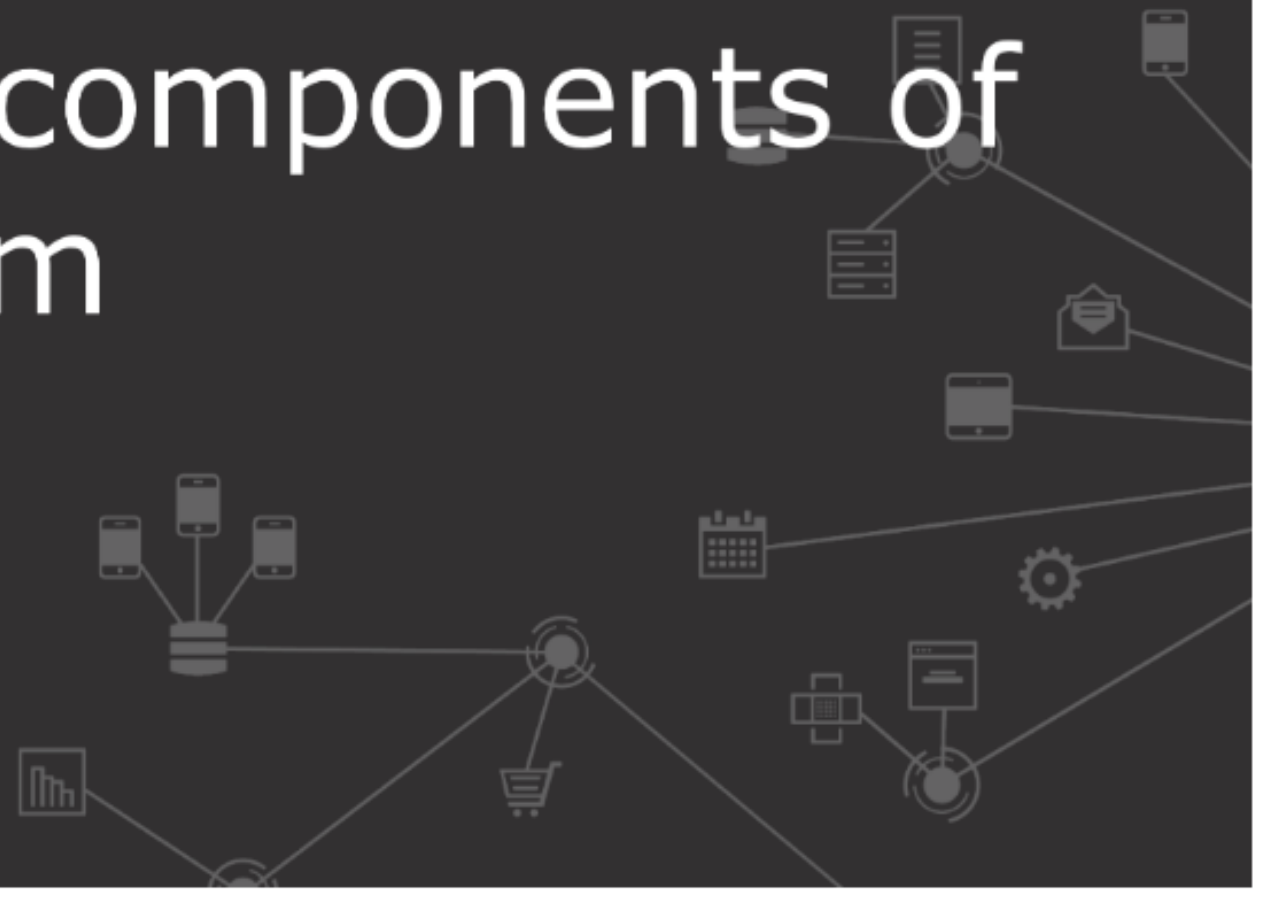
MuleSoft has a blueprint for you to follow

	Plan for success	Establish the foundation	Build to scale	Measure impact
Business outcomes	Agree on business outcomes and KPIs Develop the overall success plan	Monitor and manage	Refresh the success plan	Measure business outcomes
Technology delivery	Define Anypoint platform vision and roadmap Design Anypoint platform architecture and implementation plan	Deploy Anypoint Platform	Refine and scale Anypoint Platform	Measure Anypoint platform KPIs
	Prioritize IT projects and quick wins Staff and onboard the project teams	Define reference architecture Launch initial projects and quick wins	Onboard additional project teams Launch additional projects	Measure project KPIs
Org enablement	Assess integration capabilities Establish the C4E operating model	Build and publish foundational assets Evangelize	Drive consumption	Measure C4E KPIs
	Onboard MuleSoft Determine the internal support operating model	Staff, train and launch team Publish support guidance and self-serve materials	Monitor Anypoint Platform	Measure support KPIs
	Agree on initial roles Train the initial team(s)	Develop the broader training plan Launch experiential learning opportunities	Update training plan	Conduct skills assessment

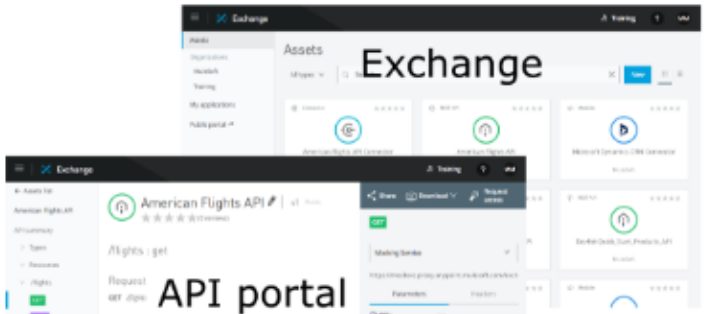
All contents © MuleSoft Inc.

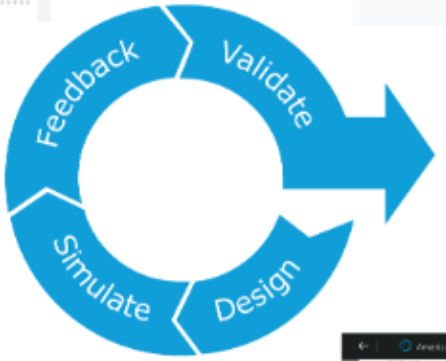
Note: For details about individual steps, contact your MuleSoft Customer Success Manager

Introducing the components of Anypoint Platform

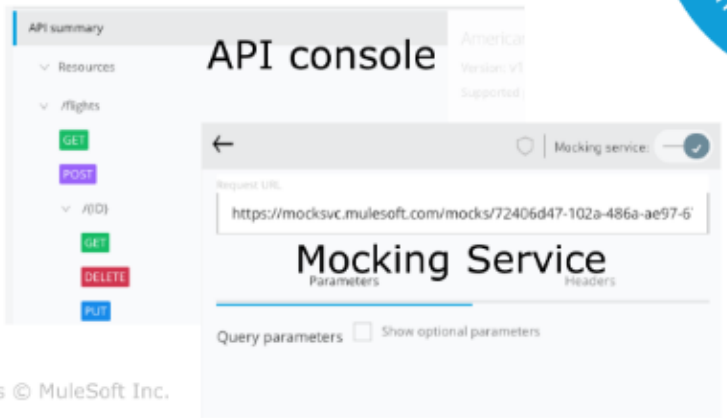


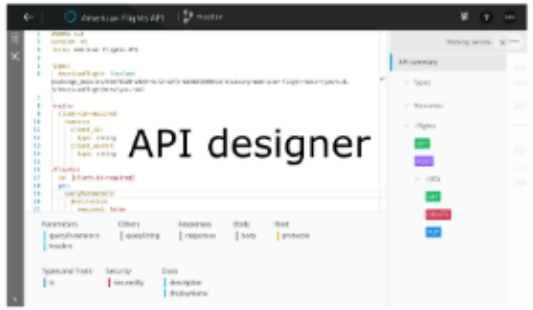
API development cycle: API specification





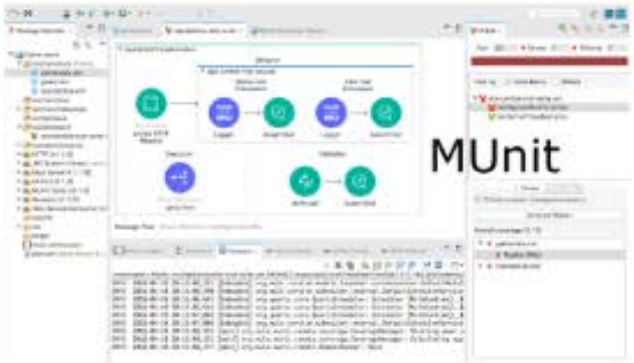
API spec (RAML)



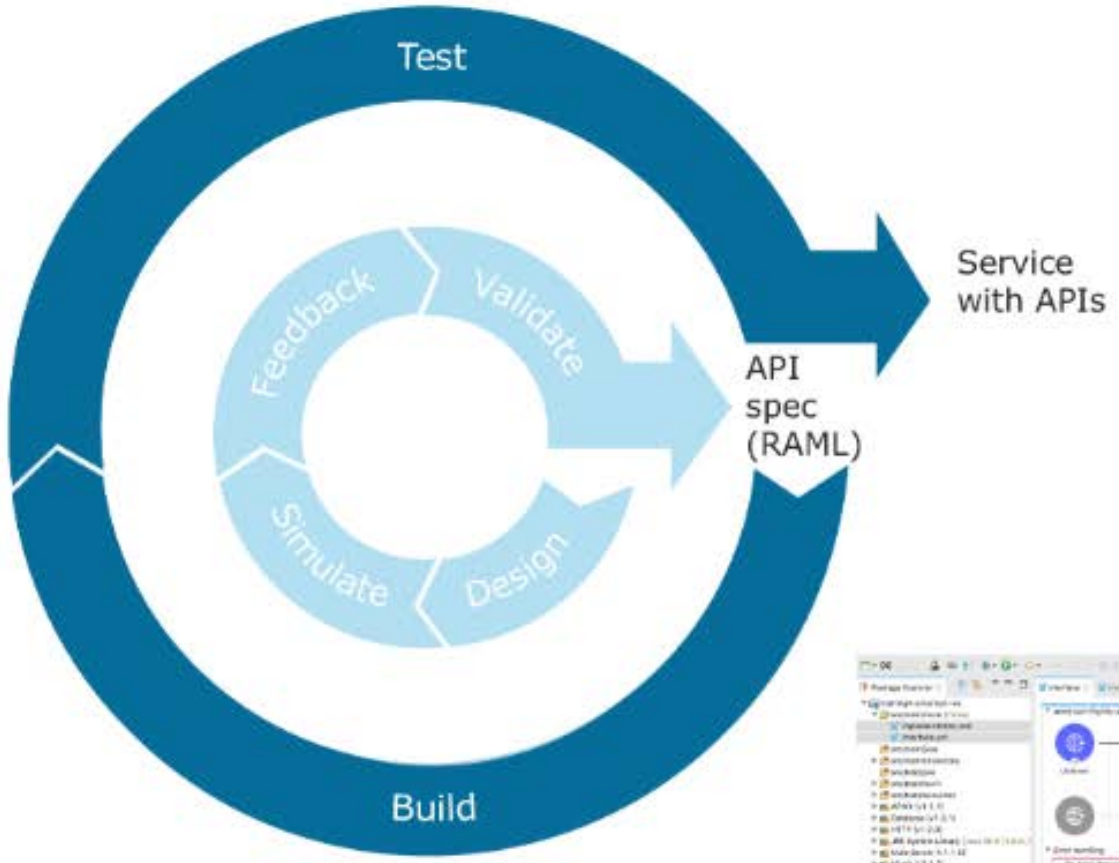


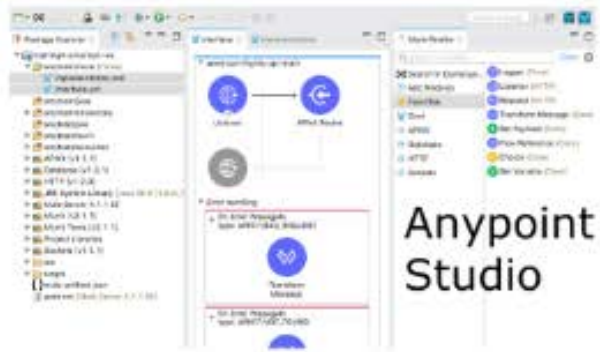
All contents © MuleSoft Inc.

API development cycle: API implementation



MUnit



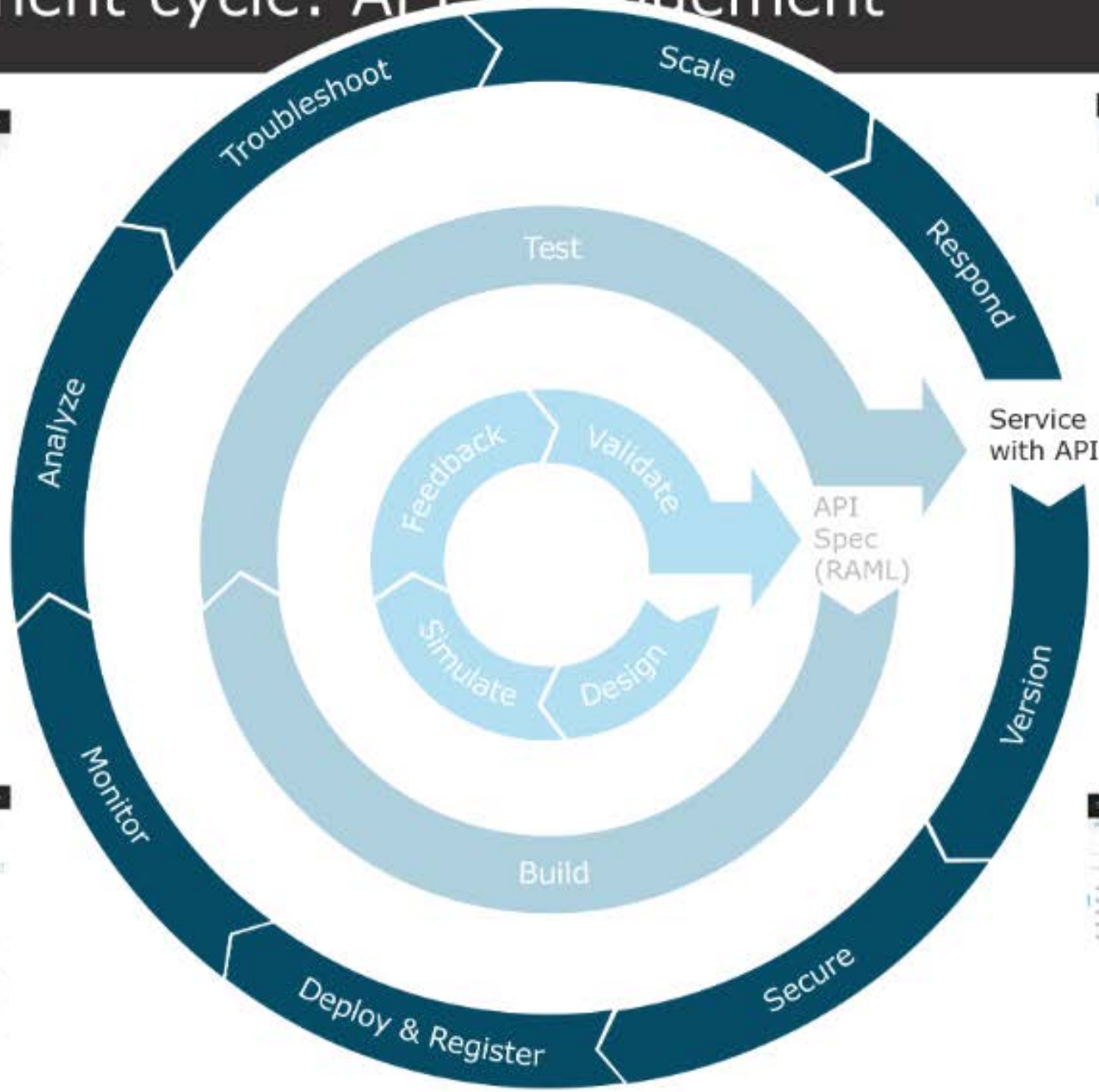


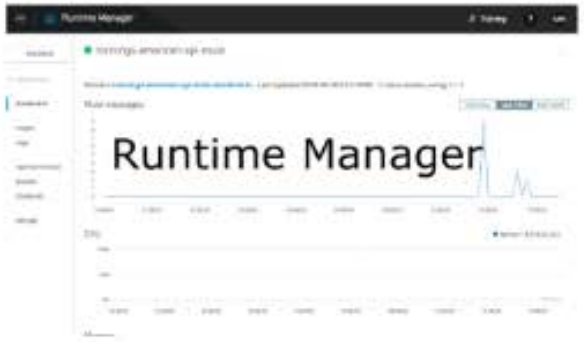
Anypoint
Studio

All contents © MuleSoft Inc.

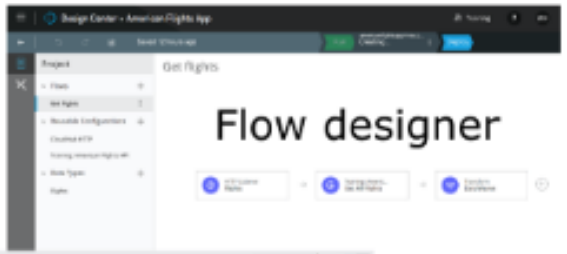
API development cycle: API management

MuleSoft

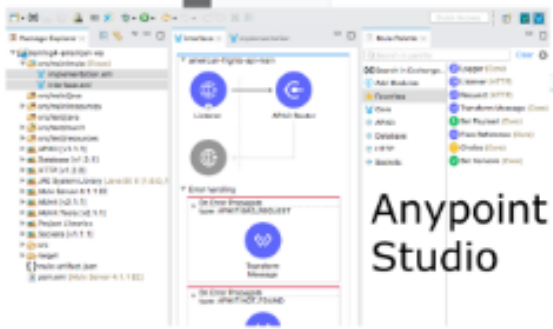




API lifecycle: Discovery and consumption



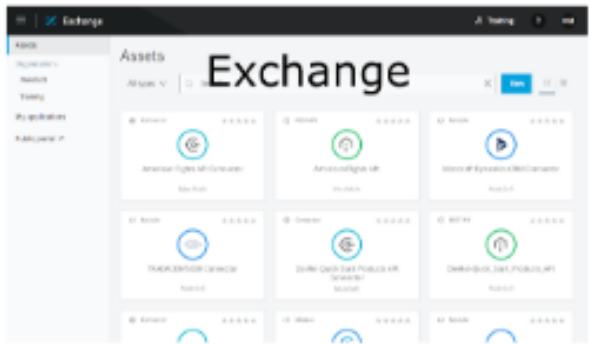
Flow designer



Anypoint Studio

Consumption

Discoverable



Exchange

Reusable assets

Feedback and usage metrics

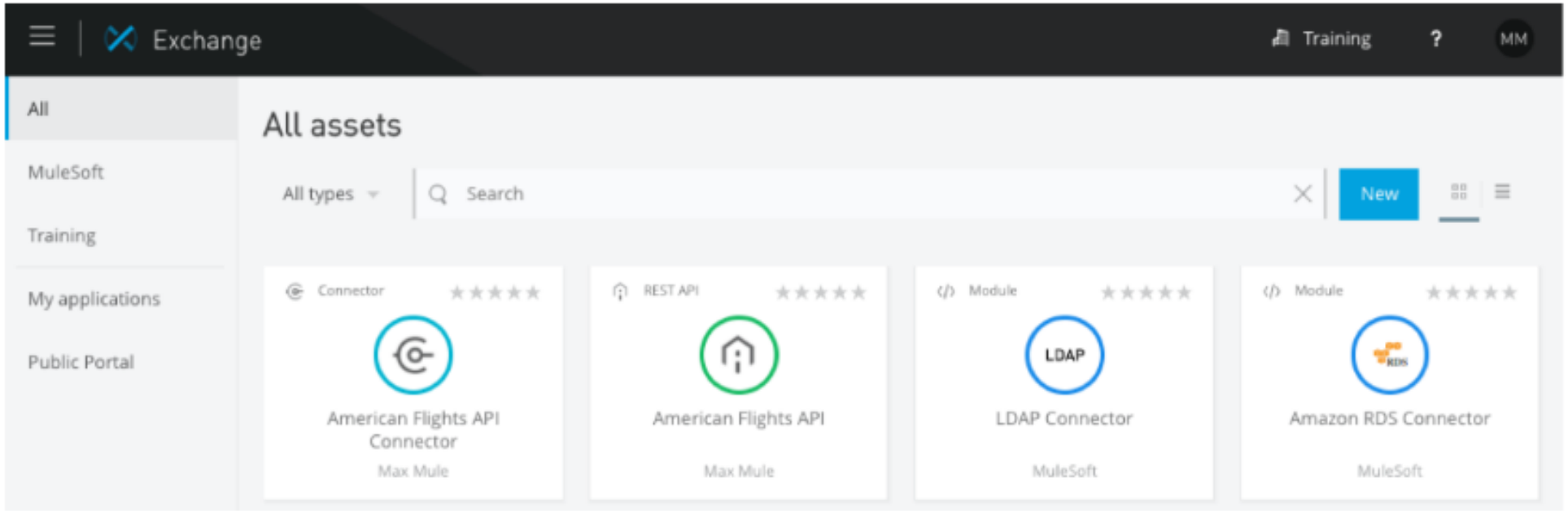
Self-service

Production

All contents © MuleSoft Inc.

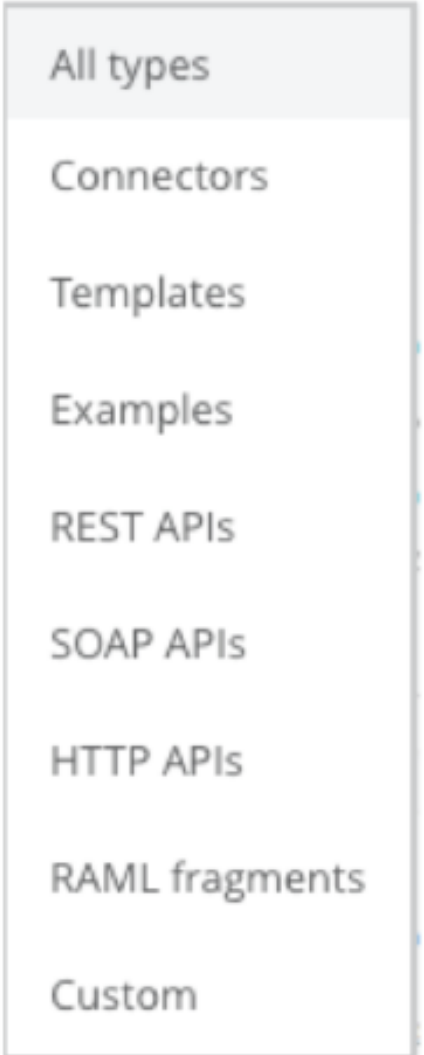
Anypoint Exchange

- A library of assets
- The central repository that is critical to the success of building an application network
- Ensures assets are published somewhere they can be discovered and reused



What does (and should) Exchange contain?

- MuleSoft-provided **public** assets available in all accounts to all users
  - You can work with MuleSoft to get APIs and connectors certified and added
- **Private** content only available to people in your org
  - Assets added by anyone in your org are added to your private Exchange
- Your organization should populate it to contain everything you need to build your integration projects
  - Including APIs, connectors, diagrams, videos, links, and more



All contents © MuleSoft Inc.

REST APIs and API portals in Anypoint Exchange

- When a REST API is added to Exchange, an **API portal** is automatically created for it

- An API portal has
  - Auto-generated **API documentation**
  - An **API console** for consuming and testing APIs
  - An **automatically generated API endpoint** that uses a **mocking service** to allow the API to be tested without having to implement it

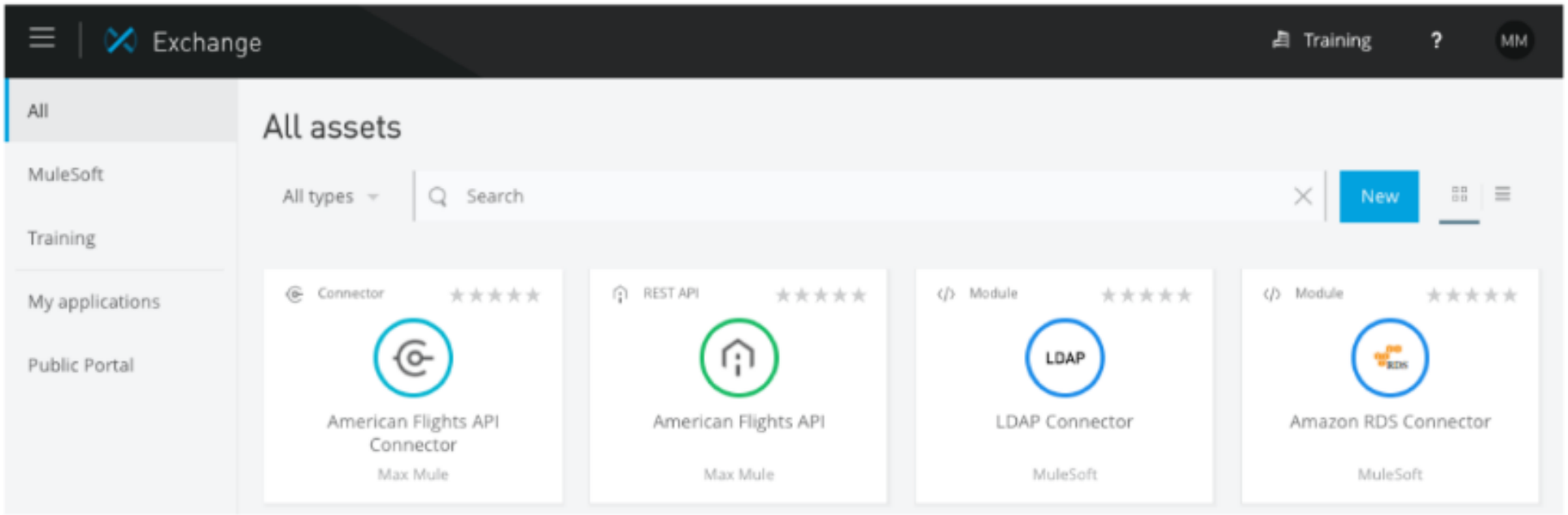
- API portals can be shared with both internal and external users

- In the last module, you used a public API portal created from Anypoint Exchange for a private organization (Muletraining)

All contents © MuleSoft Inc.

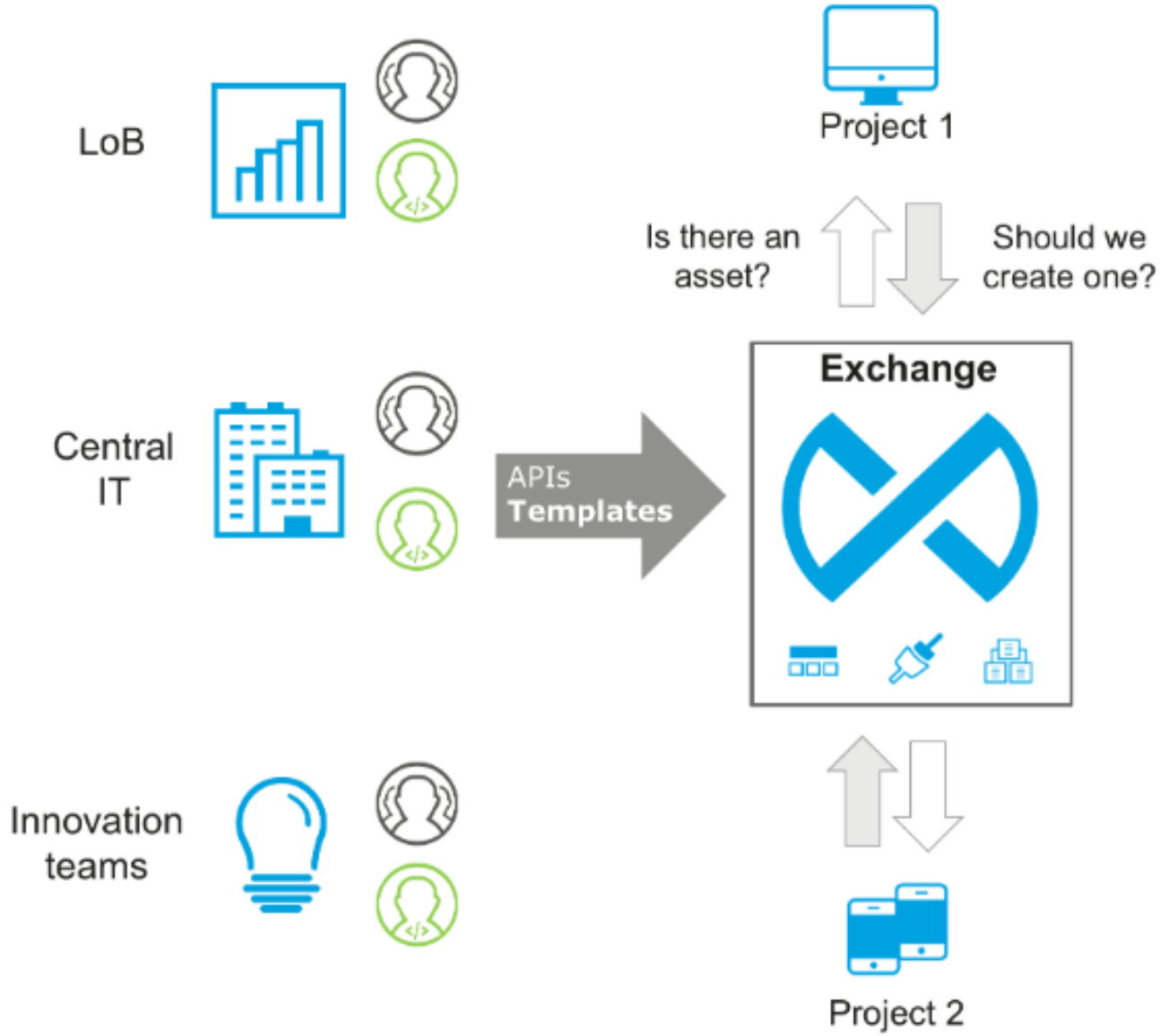
REST connectors in Anypoint Exchange

- When a RAML 1.0 API specification is added to Exchange, a **connector** is automatically created for it
  - The connector can be use in Mule applications to make calls to that API
  - REST Connect is the name of the technology that performs this conversion



All contents © MuleSoft Inc.

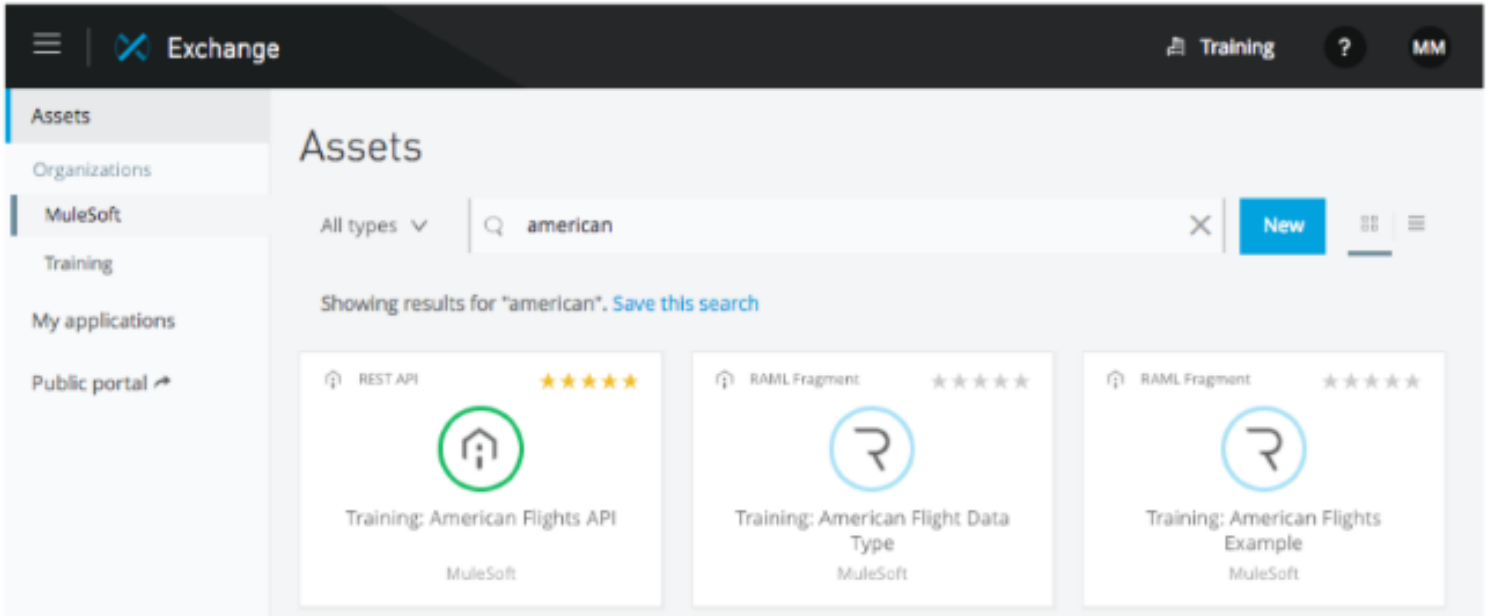
Using Exchange: Success of C4E in action



All contents © MuleSoft Inc.

Walkthrough 2-1: Explore Anypoint Platform and Anypoint Exchange

- Explore Anypoint Platform
- Browse Anypoint Exchange
- Review an API portal for a REST API in Exchange
- Discover and make calls to the Training: American Flights API in the public Exchange

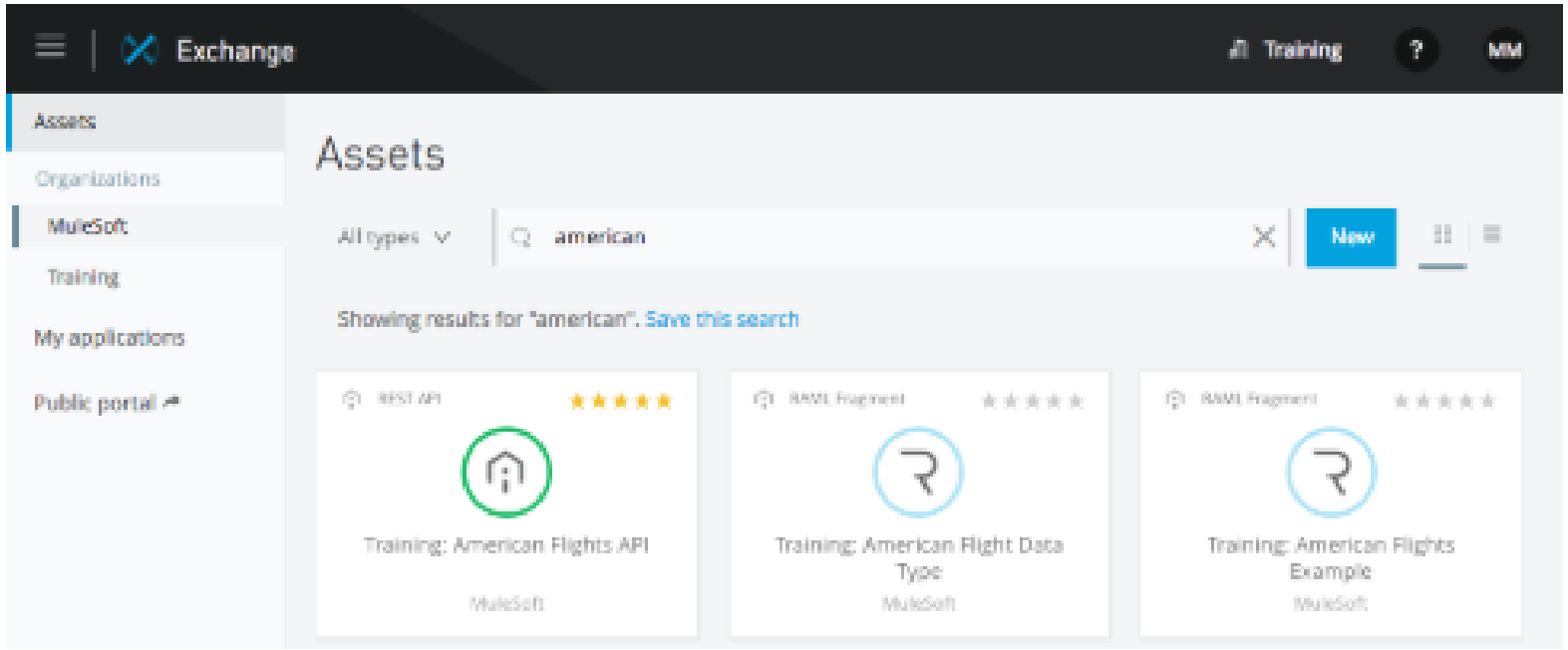


All contents © MuleSoft Inc.

Walkthrough 2-1: Explore Anypoint Platform and Anypoint Exchange

In this walkthrough, you get familiar Anypoint Platform. You will:

- Explore Anypoint Platform.
- Browse Anypoint Exchange.
- Review an API portal for a REST API in Exchange.
- Discover and make calls to the Training: American Flights API in the public Exchange.



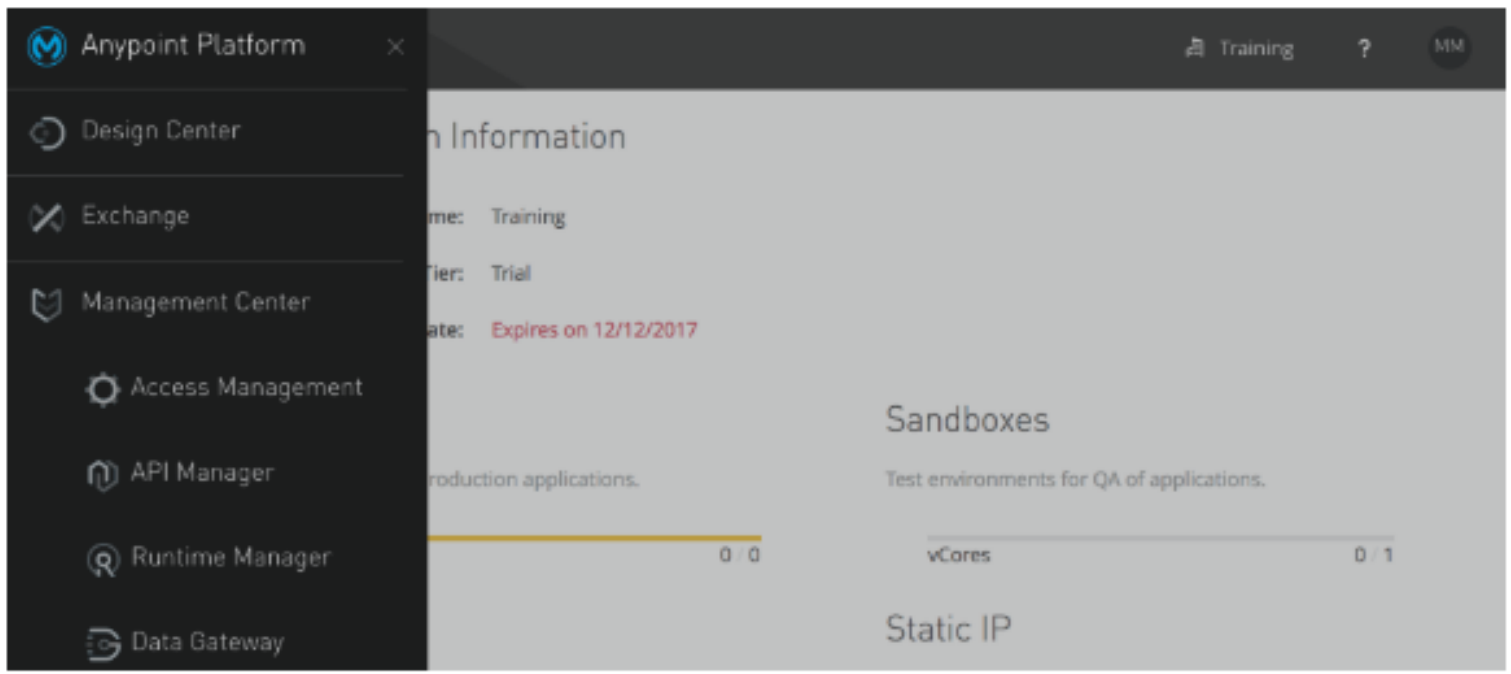
Return to Anypoint Platform

1. Return to Anypoint Platform at https://anypoint.mulesoft.com (not the public API portal you used last module!) in a web browser.

   Note: If you closed the browser window or logged out, return to https://anypoint.mulesoft.com and log in.

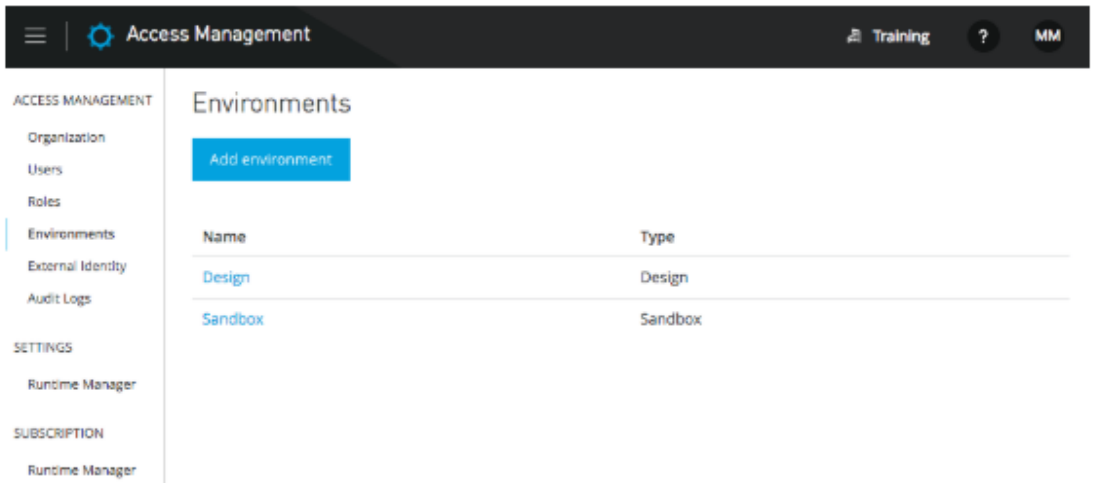
2. Click the menu button located in the upper-left in the main menu bar.
3. In the menu that appears, select Anypoint Platform; this will return you to the home page.

   Note: This will be called the main menu from now on.

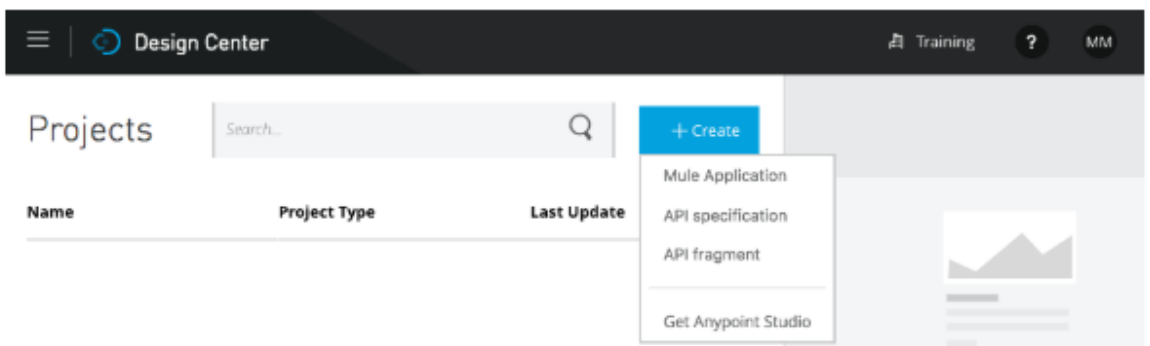


Explore Anypoint Platform

4. In the main menu, select Access Management.
5. In the left-side navigation, select Users.
6. In the left-side navigation, select Environments.

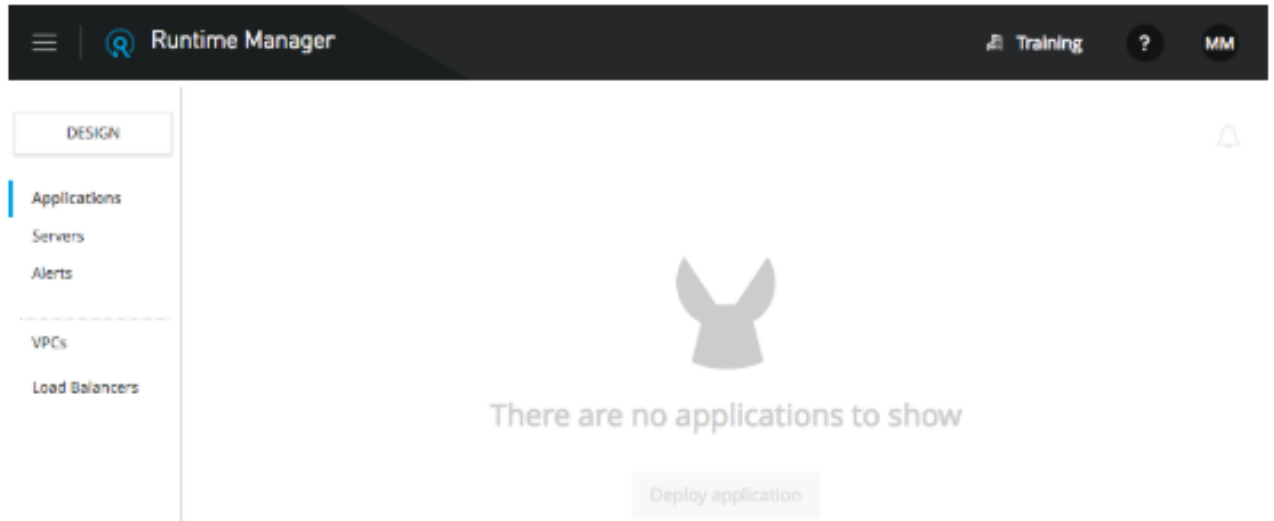


7. In the main menu, select Design Center.
8. Click the Create button and look at the options in the drop-down menu.

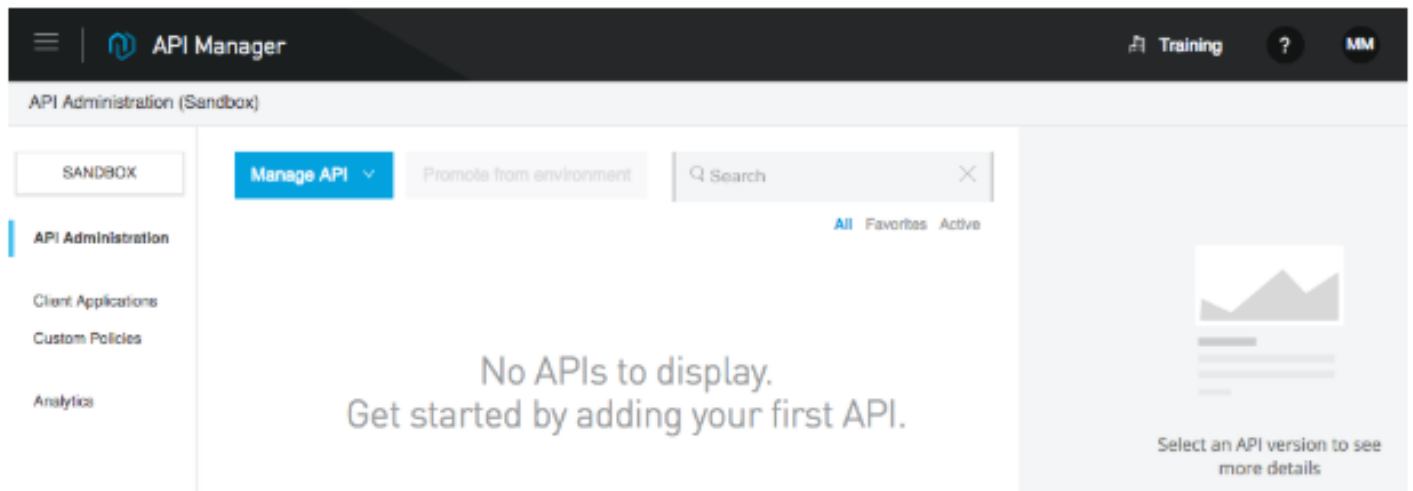


9. In the main menu, select Runtime Manager.

10. If you get a Choose environment page, select Design.

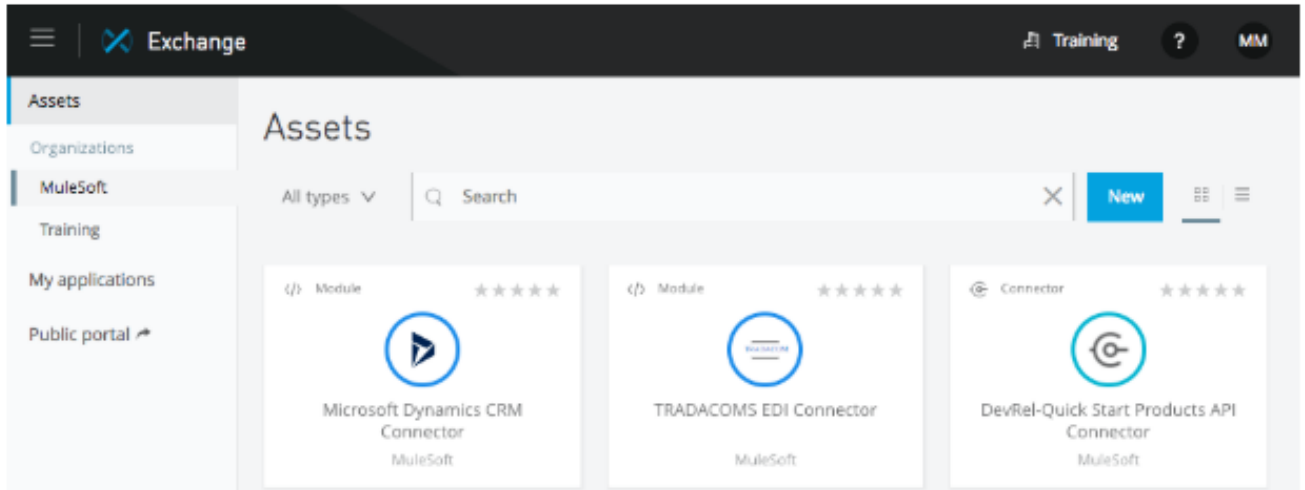


11. In the main menu, select API Manager.

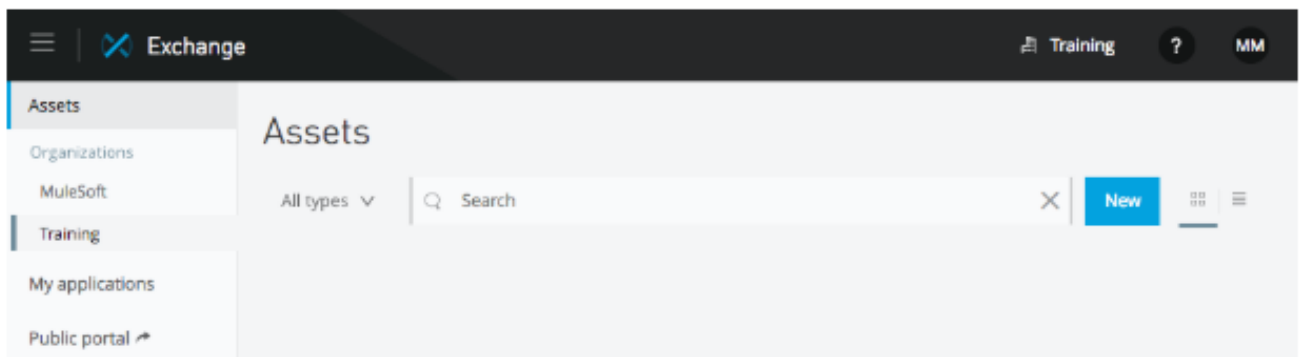


Explore Anypoint Exchange

12. In the main menu, select Exchange.
13. In the left-side navigation, select MuleSoft; you should see all the content in the public Exchange.

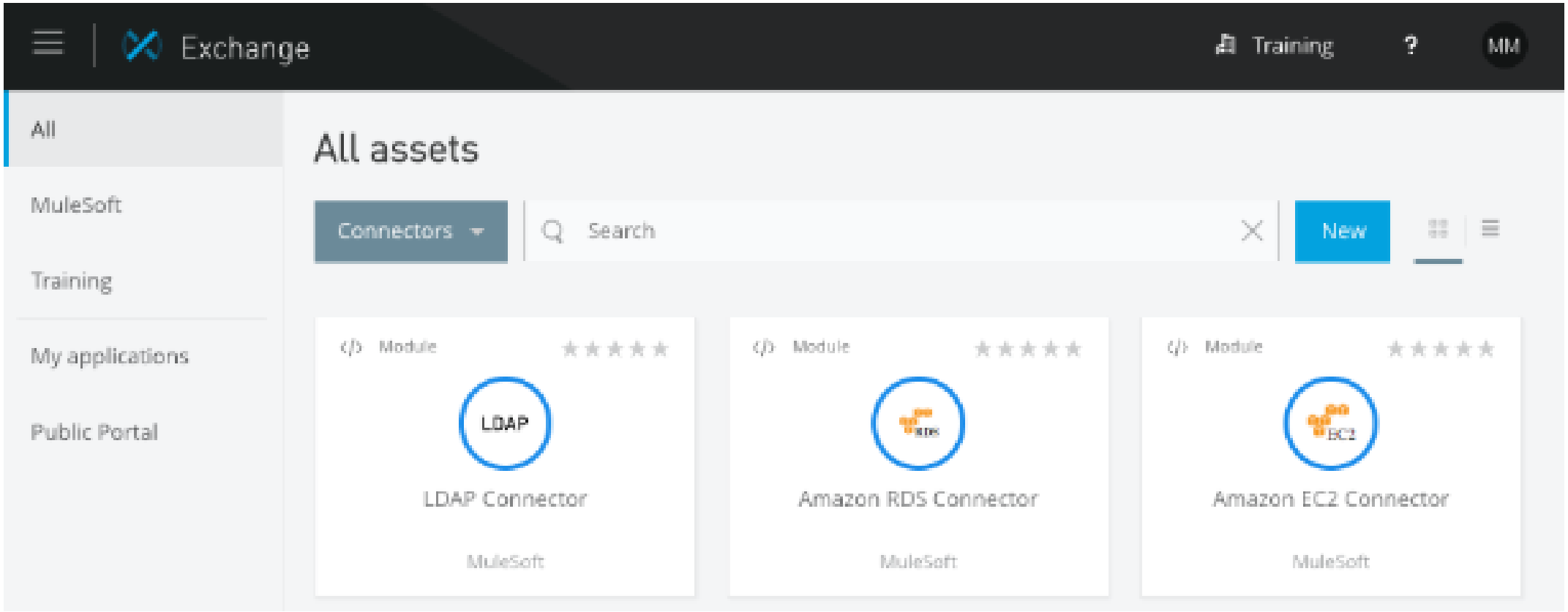


14. In the left-side navigation, select the name of your organization beneath MuleSoft (Training in the screenshots); you should now see only the content in your private Exchange, which is currently empty.



15. In the left-side navigation, select MuleSoft.

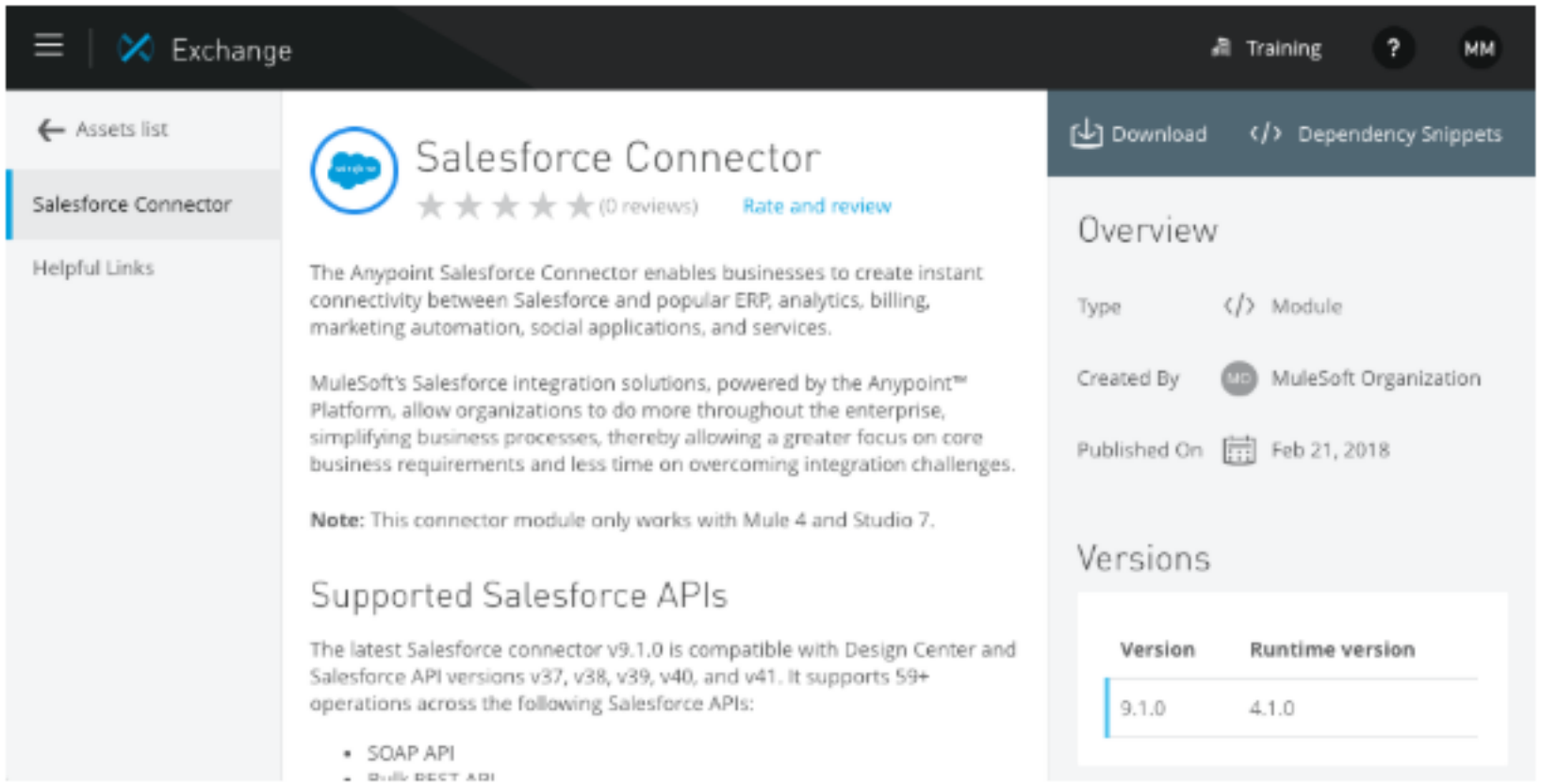
16. In the types menu, select Connectors.



17. Select one of the connectors and review its information.
18. In the left-side navigation, click the Assets list link.

19. Search for the Salesforce Connector and review its details.

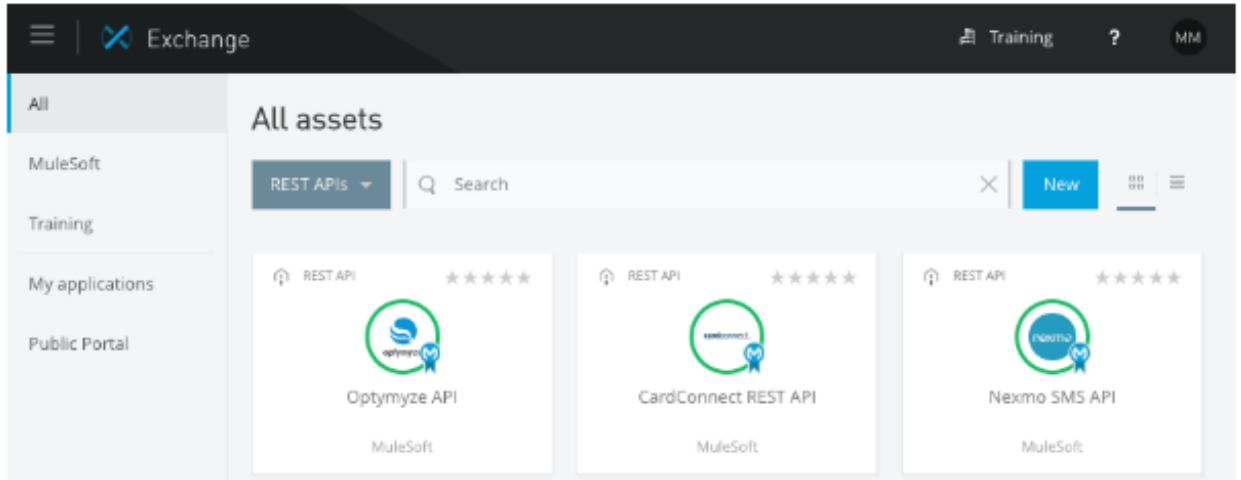
    Note: The Salesforce Connector is used in the Development Fundamentals course.



20. In the left-side navigation, click Assets list.
21. In the types menu, select Templates.
22. Remove salesforce from the search field and press Enter/Return.

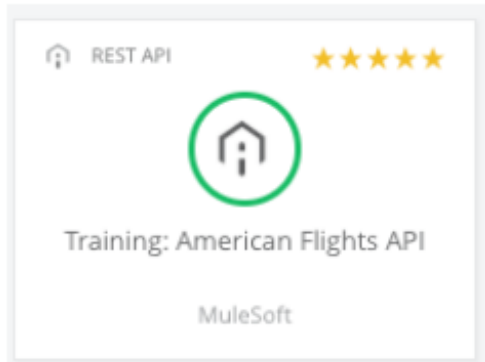
Browse REST APIs in Anypoint Exchange

23. In the types menu, select REST APIs.
24. Browse the APIs.

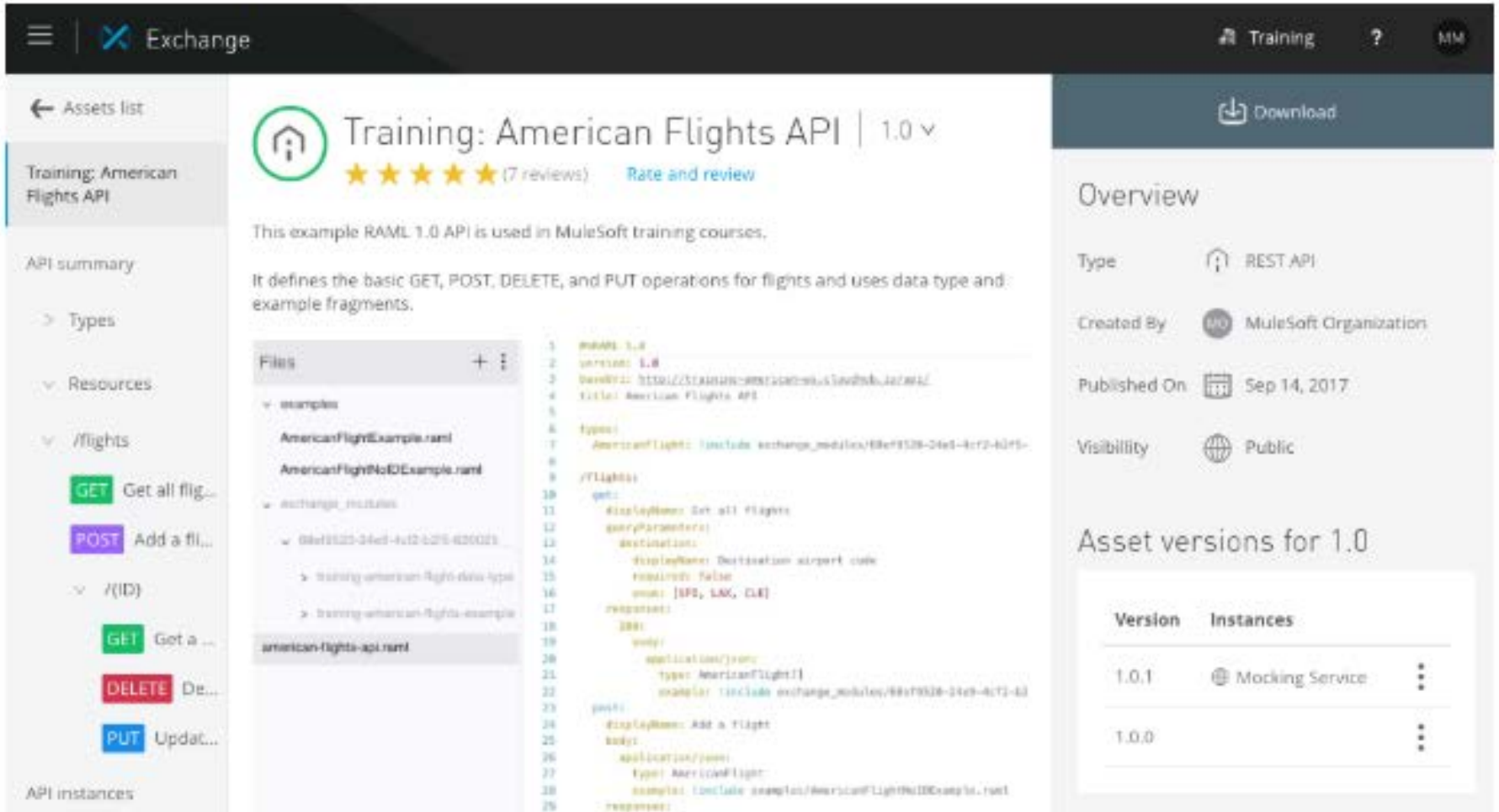


Discover and review the API portal for the Training: American Flights API

25. Locate and click the Training: American Flights API.

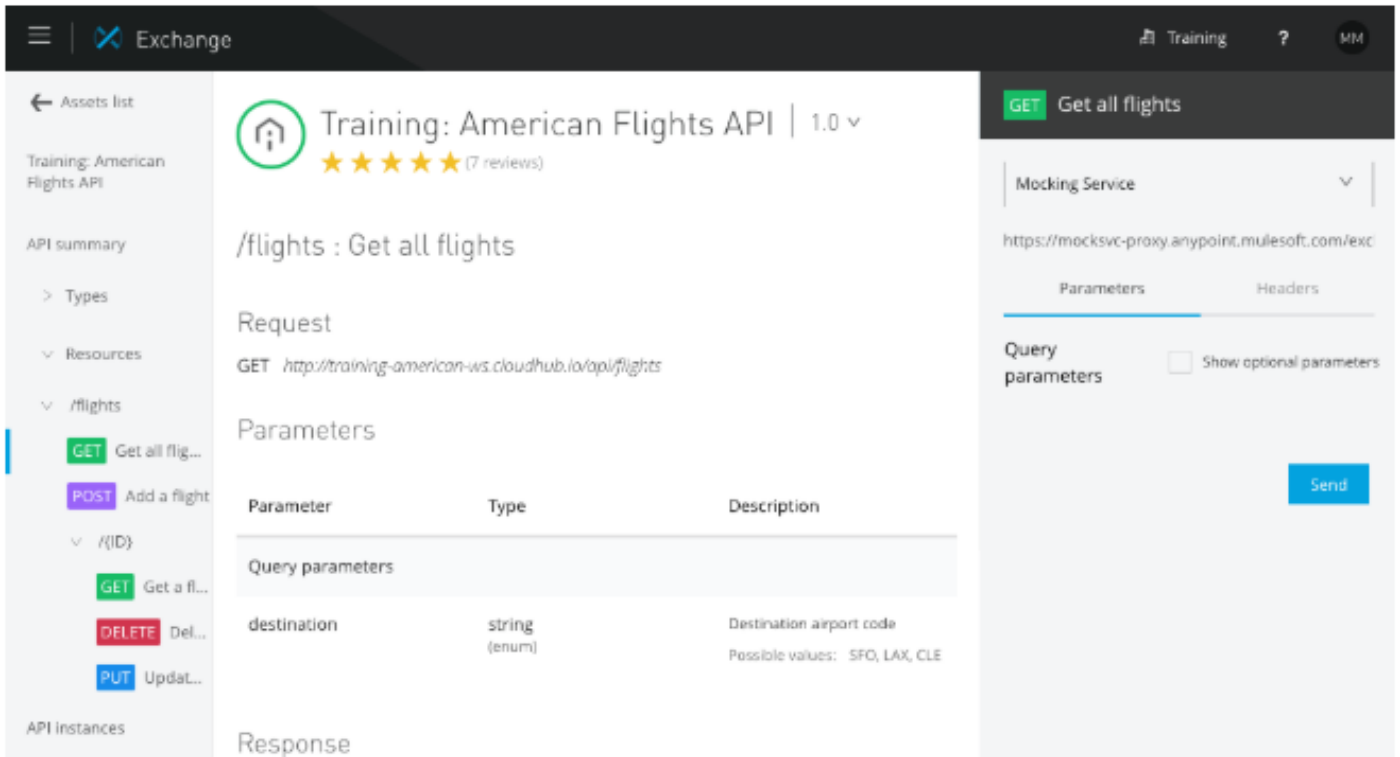


26. Review the API portal.



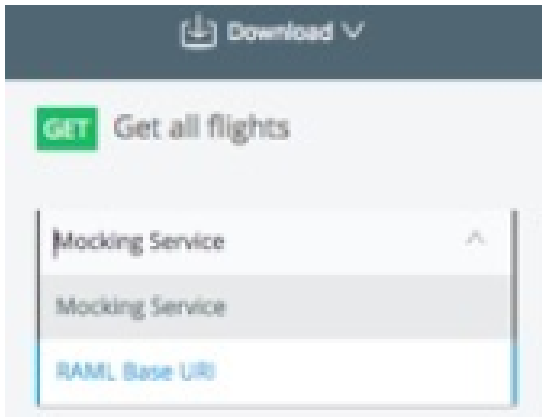
27. In the left-side navigation, expand and review the list of available resources.
28. Click the GET link for the /flights resource.

29. On the /flights: Get all flights page, review the information for the optional destination query parameter; you should see the API is similar to the one you explored in the public Muletraining portal.



Use the API console to make calls to the Training: American Flights API

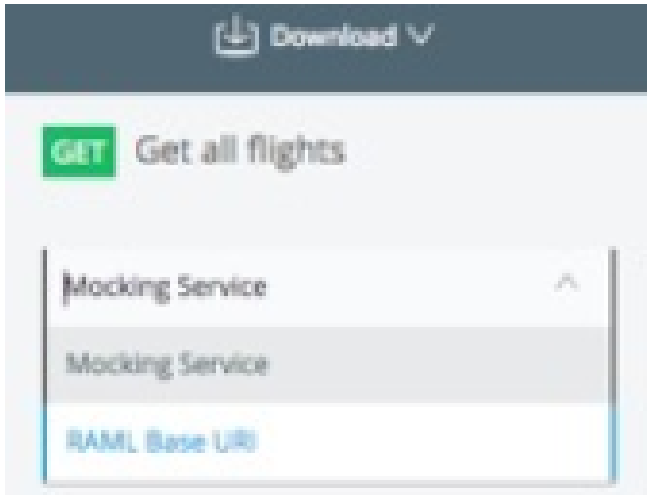
30. In the API console, review the options for the instances you can test.



31. Select Mocking Service.

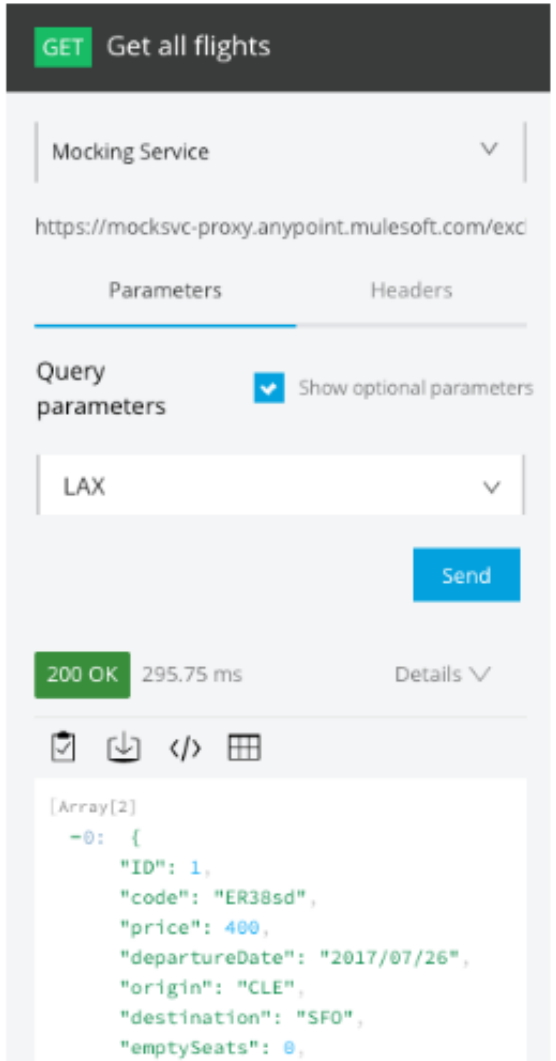
Use the API console to make calls to the Training: American Flights API

30. In the API console, review the options for the instances you can test.



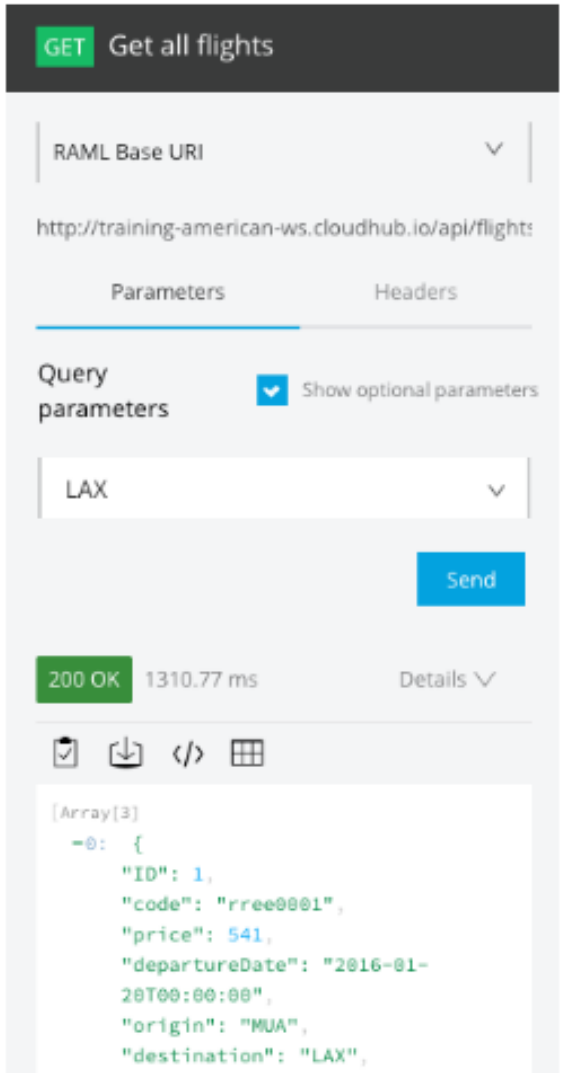
31. Select Mocking Service.

32. Select a destination and click Send; you should get the two example flights.



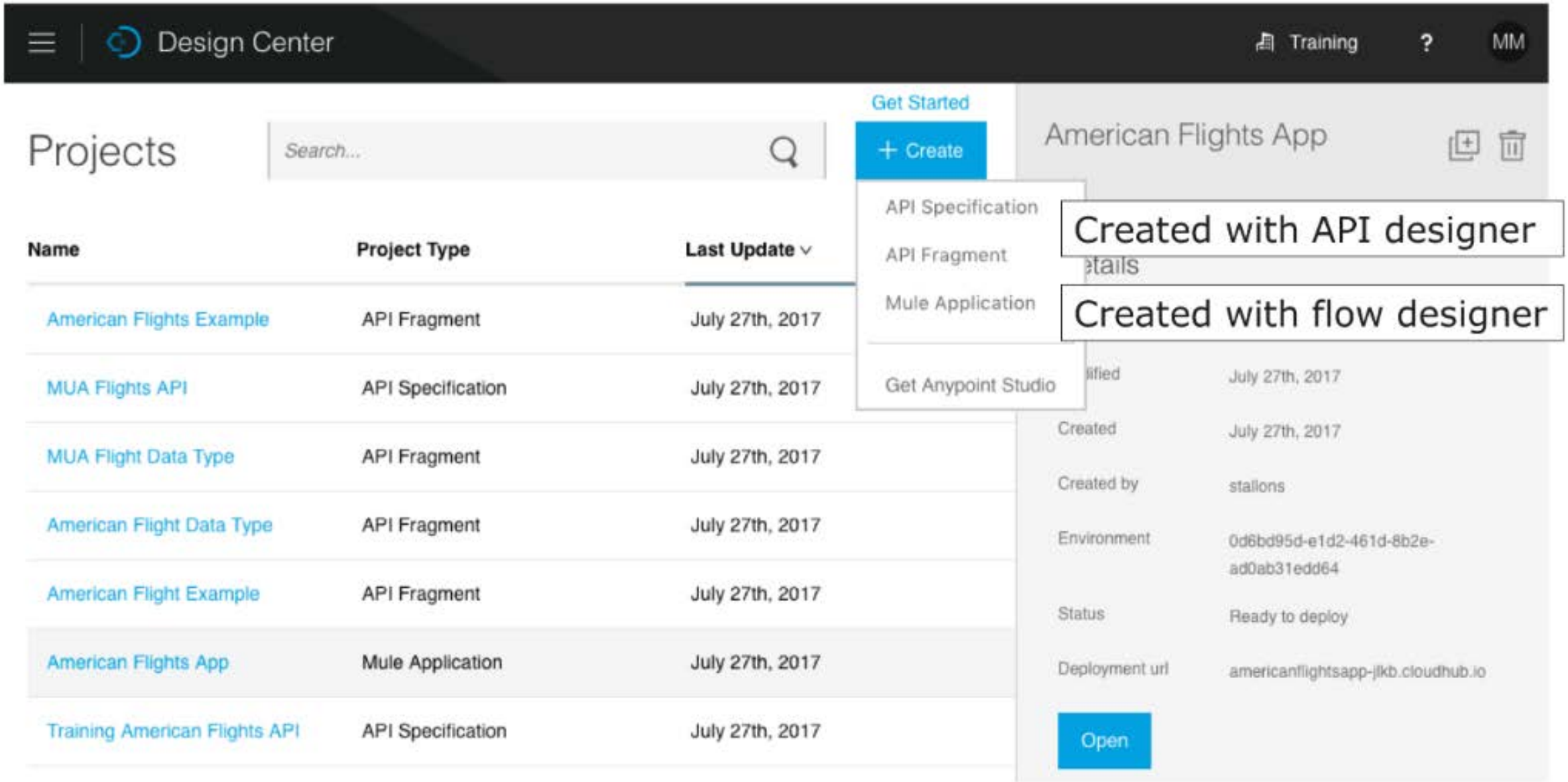
33. Change the API instance from Mocking Service to RAML Base URI.

34. Click Send again; you should get results from the actual API implementation for the destination you selected.



Building integration applications and APIs with Design Center

Design Center anatomy



Design Center applications

Application	Purpose	In this course	Additional courses
flow designer	Web app for building integration apps that connect systems and consume APIs	2 WTs	• Anypoint Platform: Flow Design
API designer	Web app for designing, documenting, and mocking APIs	Module 3	• Anypoint Platform: API Design
Anypoint Studio	Desktop IDE for implementing APIs and building integration applications	Module 4 In Fundamentals: Modules 6-13	

All contents © MuleSoft Inc.

Both flow designer and Anypoint Studio create Mule applications

- **Mule applications** can be created
  - Visually using flow designer or Anypoint Studio
  - By writing code (primarily XML) using Anypoint Studio (or other tools)
- Under the hood, Mule applications are Java applications using Spring
- Mule applications are deployed to a **Mule runtime**
  - Mule runtimes can be MuleSoft-hosted in the cloud (CloudHub) or customer-hosted in the cloud or on-prem

All contents © MuleSoft Inc.

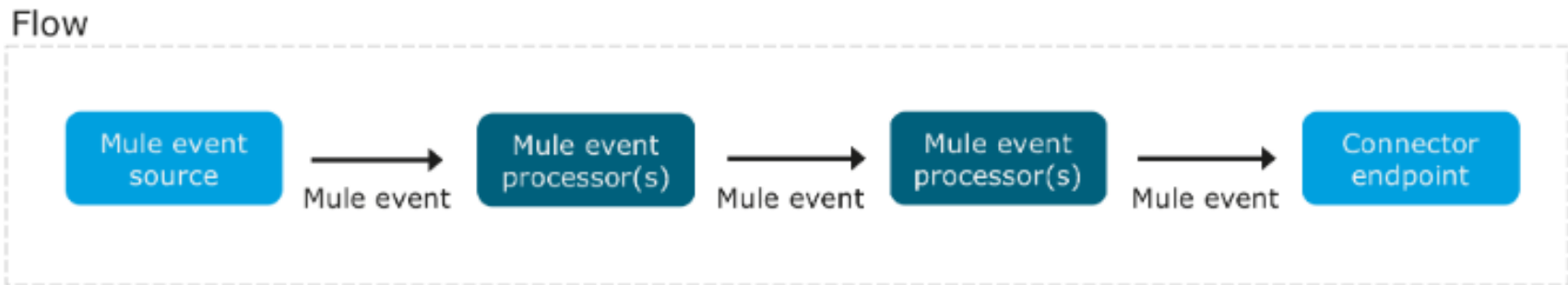
Mule is the runtime engine of Anypoint Platform

- **A lightweight Java-based enterprise service bus (ESB) and integration platform** that allows developers to connect apps together quickly and easily, enabling them to exchange data
  - Acts as a transit system for carrying data between apps (the Mule)
  - Can connect all systems including web services, JMS, JDBC, HTTP, & more
- **Decouples point-to-point integrations** by having all (non-Mule) apps talk to the bus (to a Mule runtime) instead of directly to each other
- **Can be deployed anywhere**, can integrate and orchestrate events in real time or in batch, and has universal connectivity
- **Enforces policies for API governance**

All contents © MuleSoft Inc.

Mule 4 applications and flows

- Mule applications receive events, process them, and route them to other endpoints
- **Mule applications** accept and process a **Mule event** through a series of **Mule event processors** plugged together in a **flow**



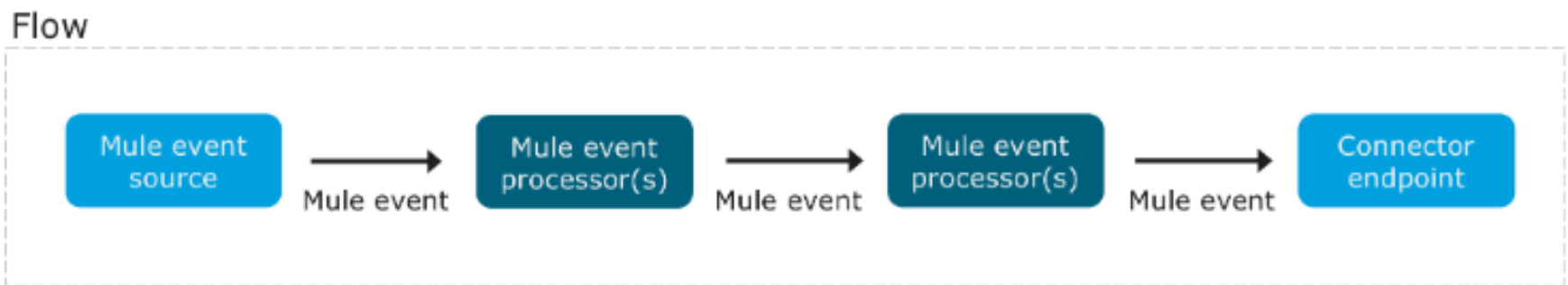
- An application can consist of
  - A single flow
  - Multiple flows
  - Multiple flows connected together

All contents © MuleSoft Inc.

What's in a typical Mule 4 flow?

- A **Mule event source** that initiates the execution of the flow
  - Can be triggered by an event like
    - A consumer request from a mobile device
    - A change to data in a database
    - The creation of a new customer ID in a SaaS application

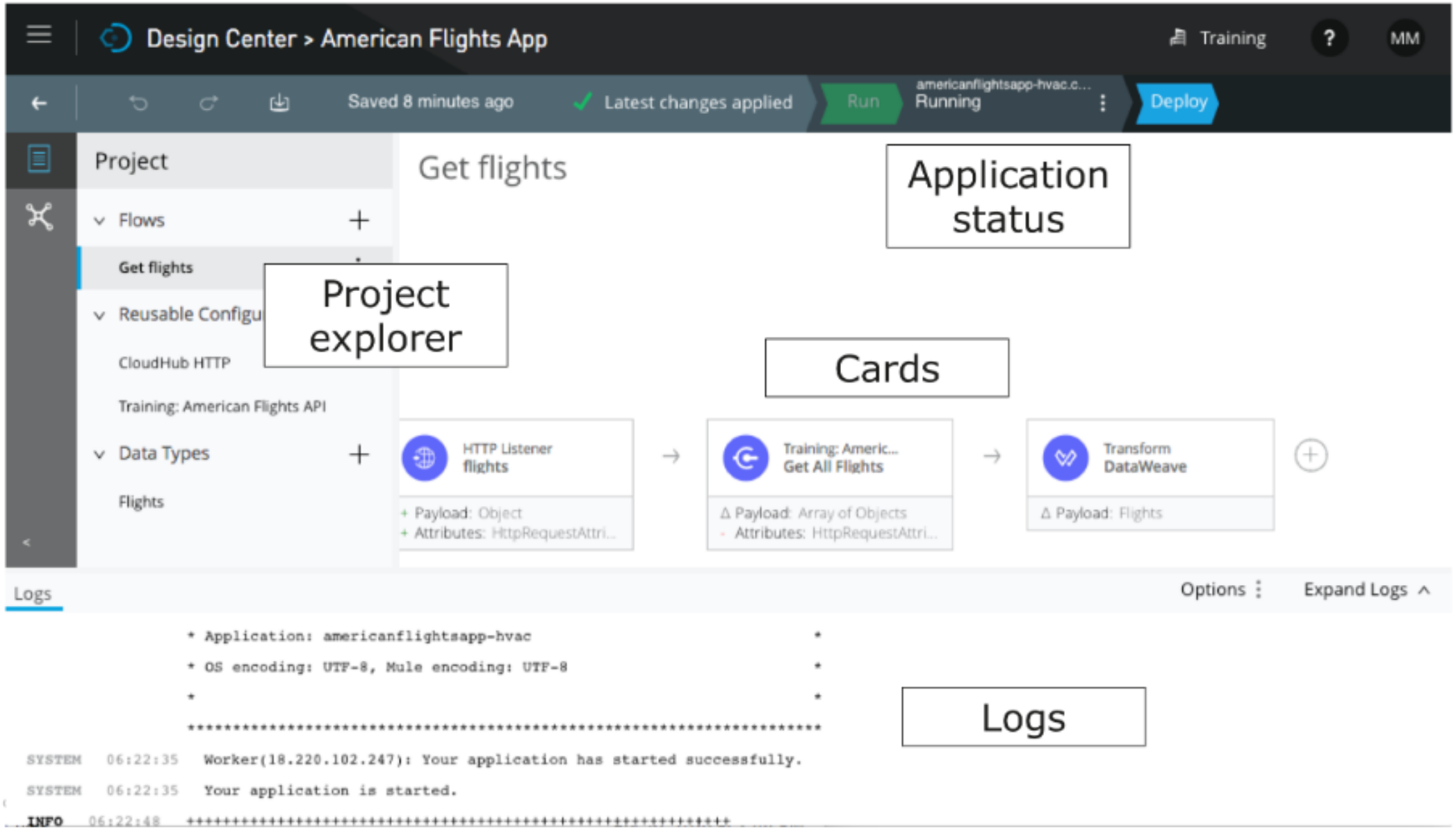
- **Mule event processors** that transform, filter, enrich, and process the event and its message



All contents © MuleSoft Inc.

Creating integration applications with flow designer

Flow designer anatomy



Running flow designer applications

- When you create a Mule application project in Design Center
  - A new application is created and opened in flow designer
  - **The application is deployed to a MuleSoft-hosted Mule runtime (called a CloudHub worker) in the cloud and started**

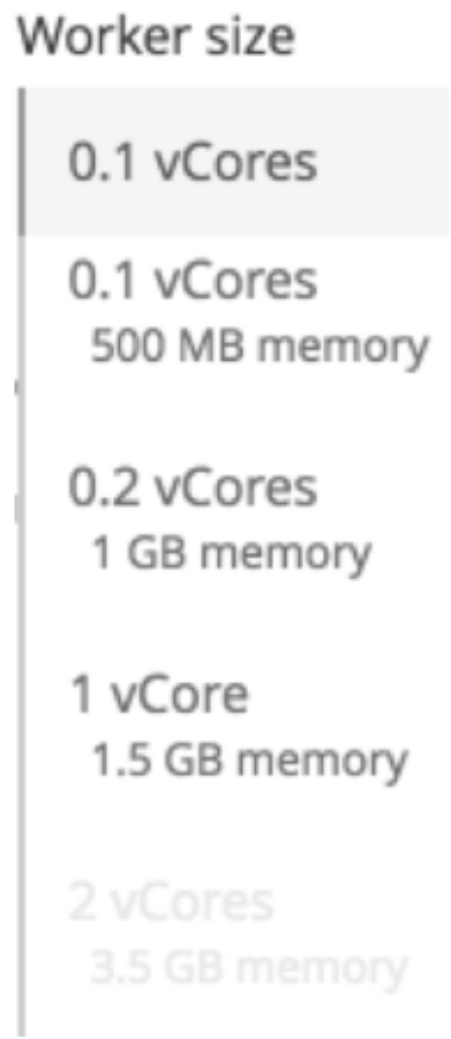
MuleSoft-hosted runtime

- When you make changes to the application in flow designer and are ready to test it
  - You need to run the application again – which updates the application deployed to the worker

All contents © MuleSoft Inc.

CloudHub workers

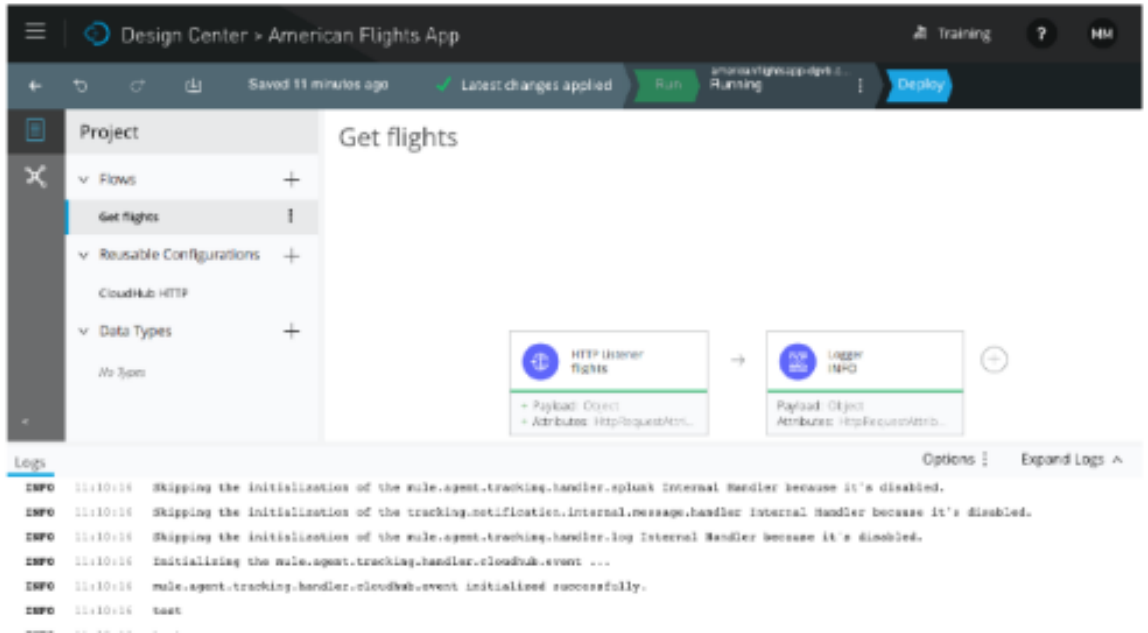
- **A worker is a dedicated instance of Mule that runs an app**
- Each worker
  - Runs in a separate container from every other application
  - Is deployed and monitored independently
  - Runs in a specific worker cloud in a region of the world
- Workers can have a different memory capacity and processing power
  - Apps can be scaled vertically by changing the worker size
  - Apps can be scaled horizontally by adding multiple workers
- There are workers in different environments
  - Design (for flow designer apps only), Sandbox, Production..
  - Apps can be promoted from one environment to another



All contents © MuleSoft Inc.

Walkthrough 2-2: Create a Mule application with flow designer

- Create a new Mule application project in Design Center
- Create an HTTP trigger for a flow in the application
- Add a Logger component
- Run and test the application
- View application information in Runtime Manager

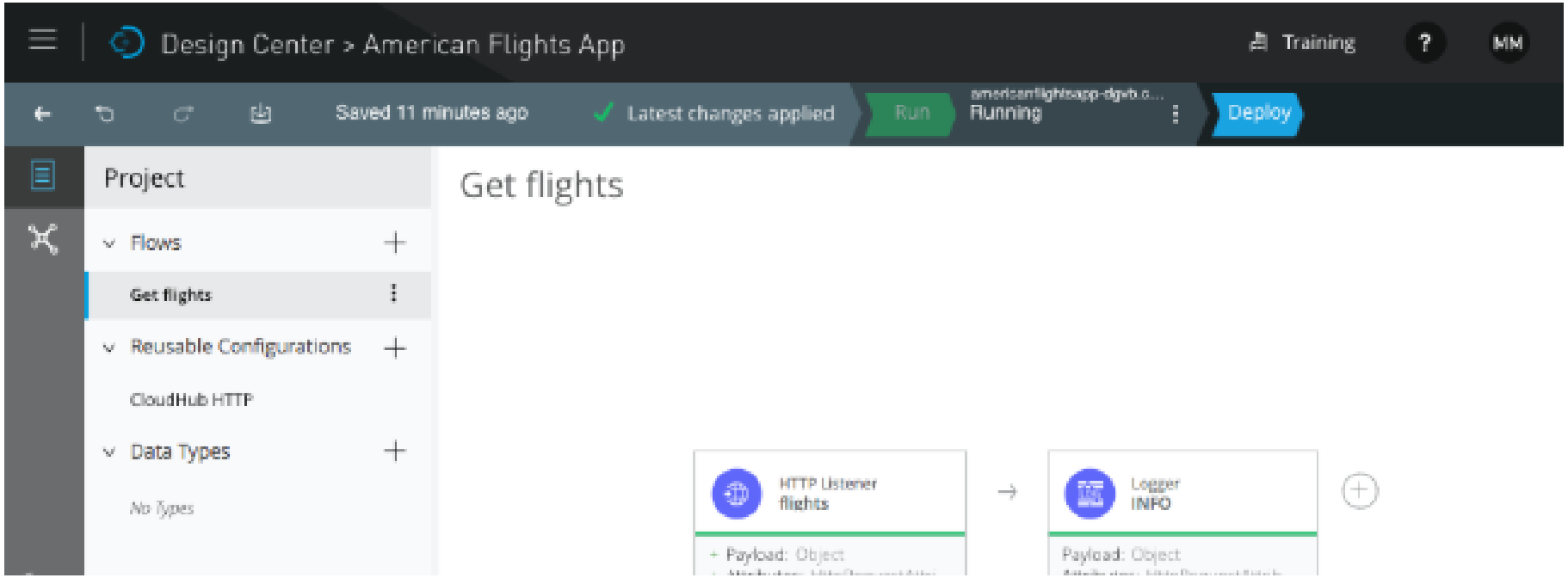


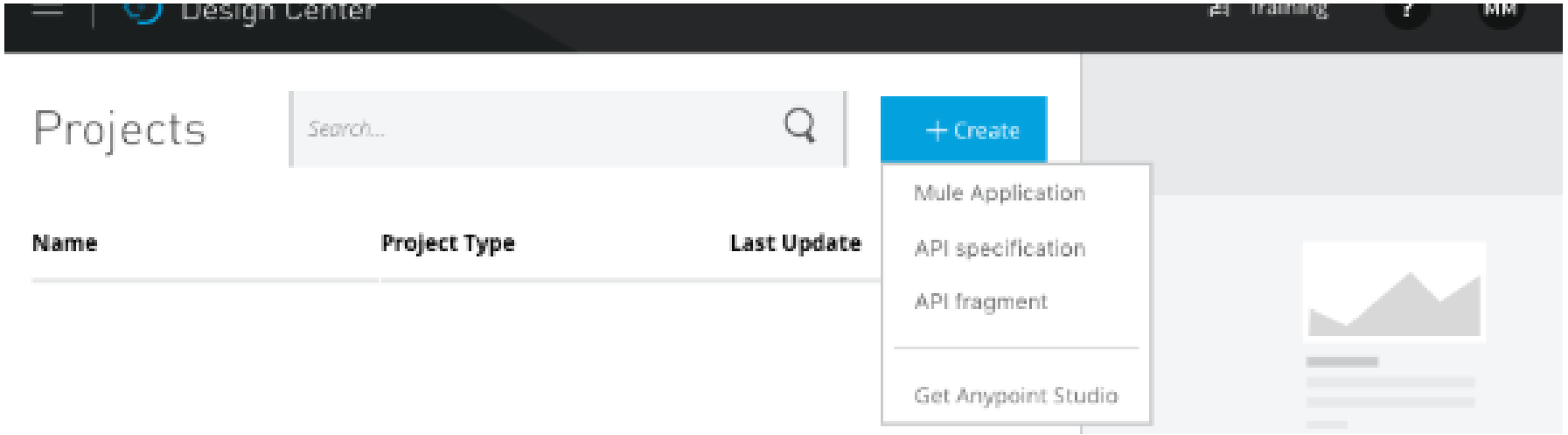
All contents © MuleSoft Inc.

Walkthrough 2-2: Create a Mule application with flow designer

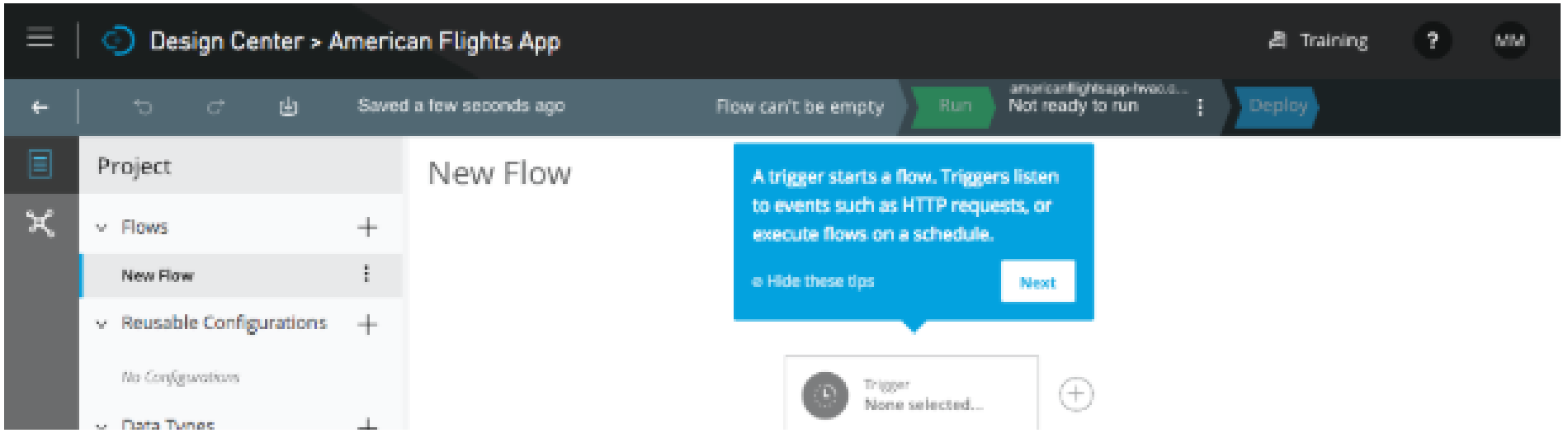
In this walkthrough, you build, run, and test a basic Mule application with flow designer. You will:

- Create a new Mule application project in Design Center.
- Create an HTTP trigger for a flow in the application.
- Add a Logger component.
- Run and test the application.
- View application information in Runtime Manager.

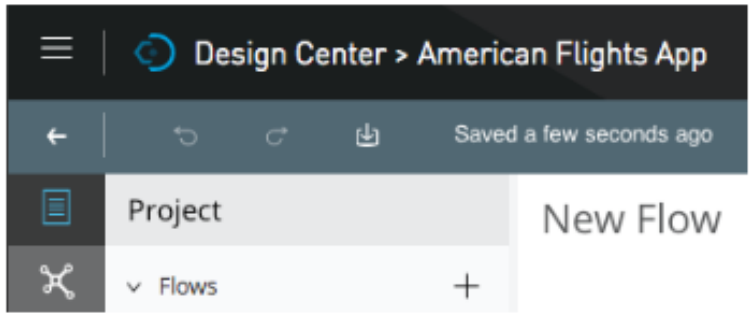




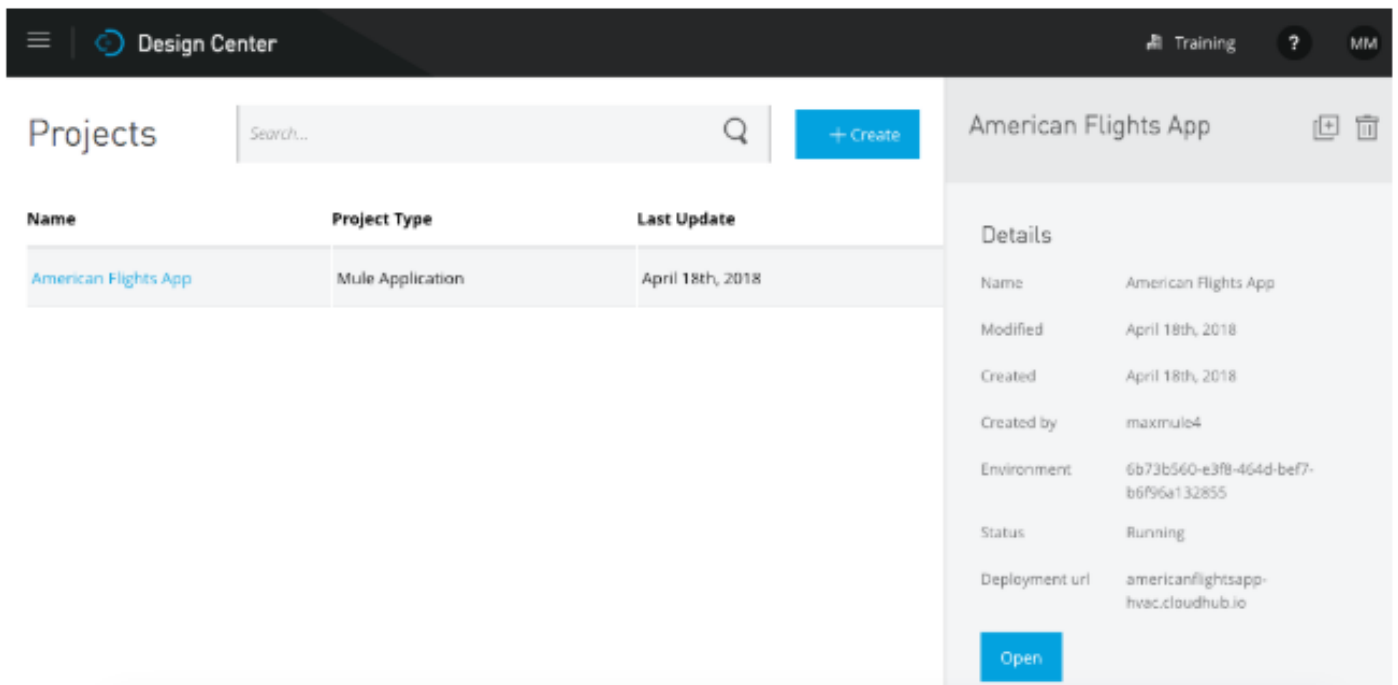
4. In the New Mule Application dialog box, set the project name to American Flights App.
5. Click Create; flow designer should open.



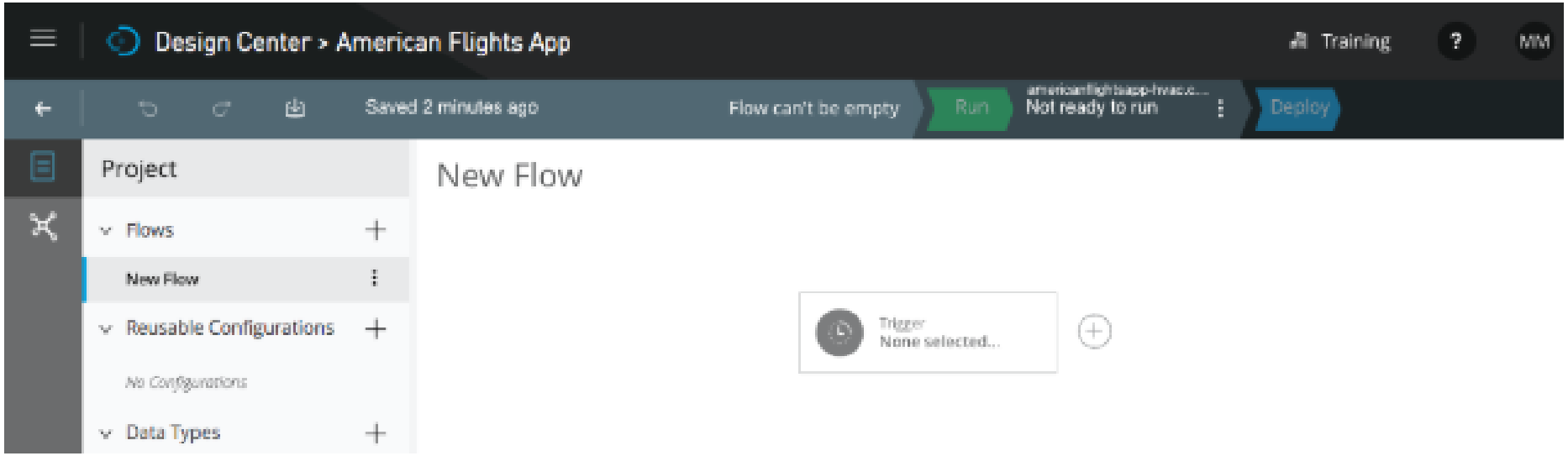
7. Click the arrow icon in the upper-left corner; you should return to the Design Center.



8. In the Design Center project list, click the row containing the American Flights App; you should see information about the project displayed on the r page.

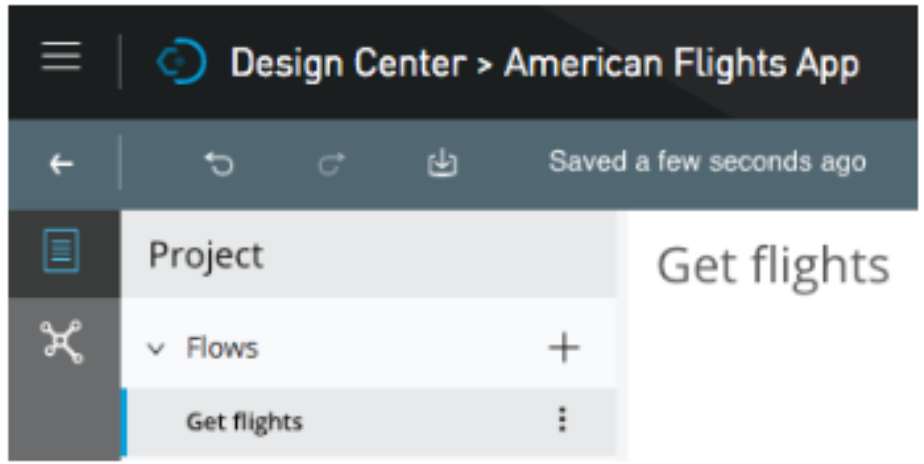


9. Click the Open button or click the American Flights App link in the project list; the project should open in flow designer.



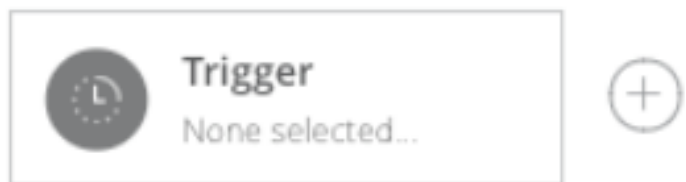
Rename the flow

10. Locate New Flow in the project explorer.
11. Click its option menu and select Rename.
12. In the Rename Flow dialog box, set the name to Get flights and click OK.

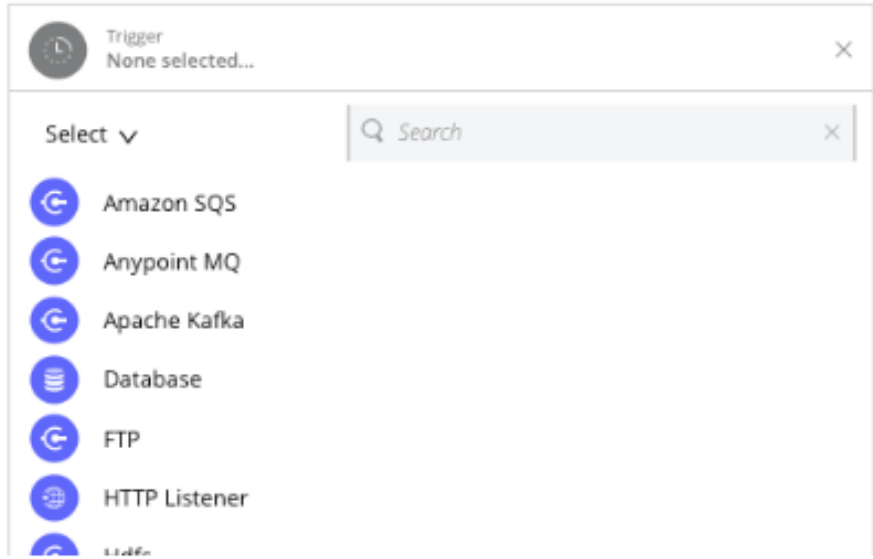


Create an HTTP trigger for a flow in the application

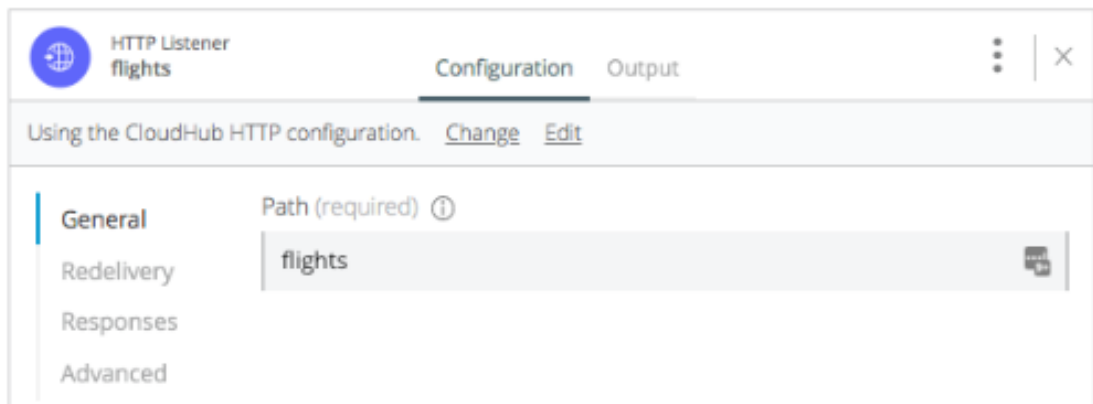
13. In flow designer, click the Trigger card.



14. In the Trigger card, select HTTP Listener.

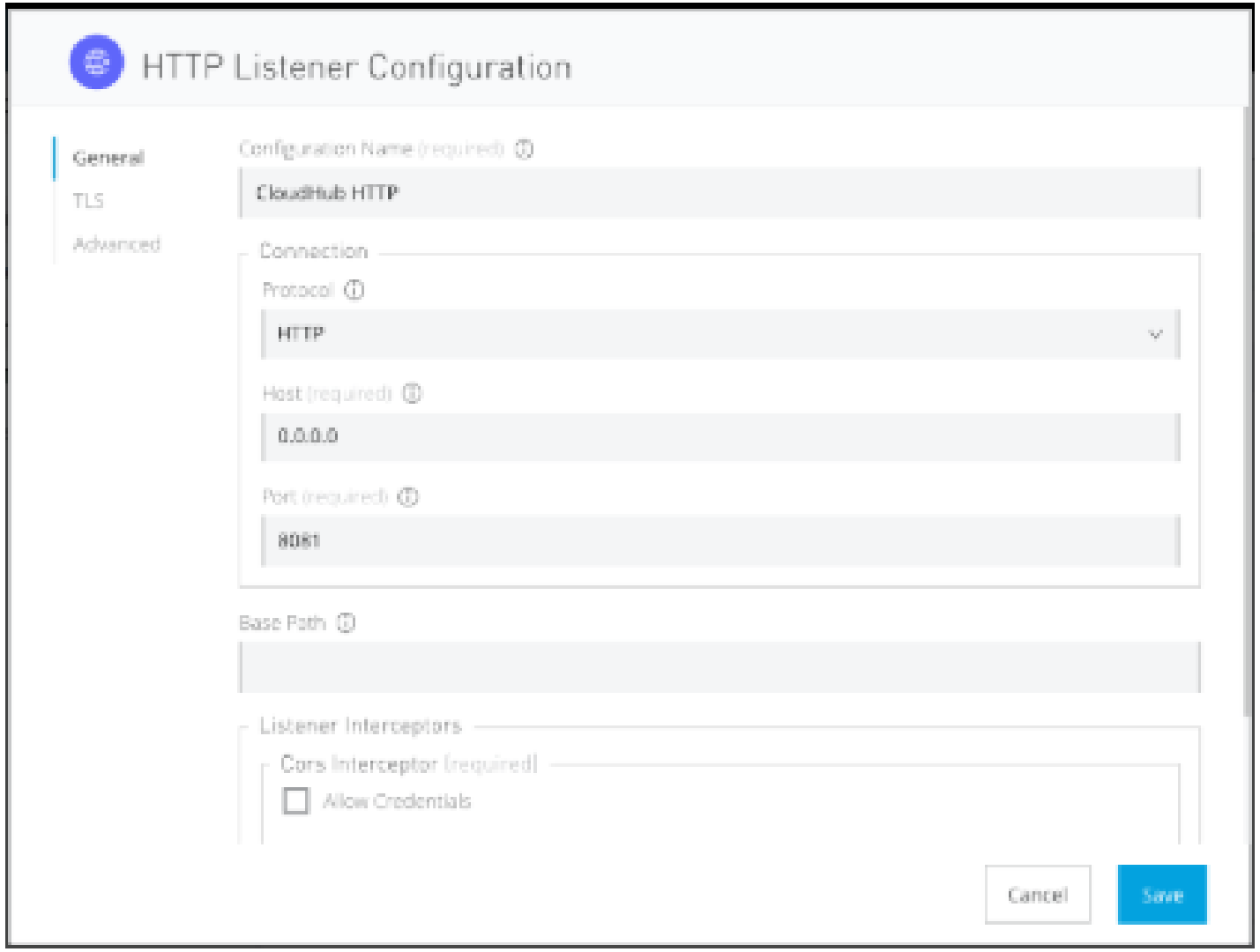


15. In the HTTP Listener dialog box, set the path to flights.

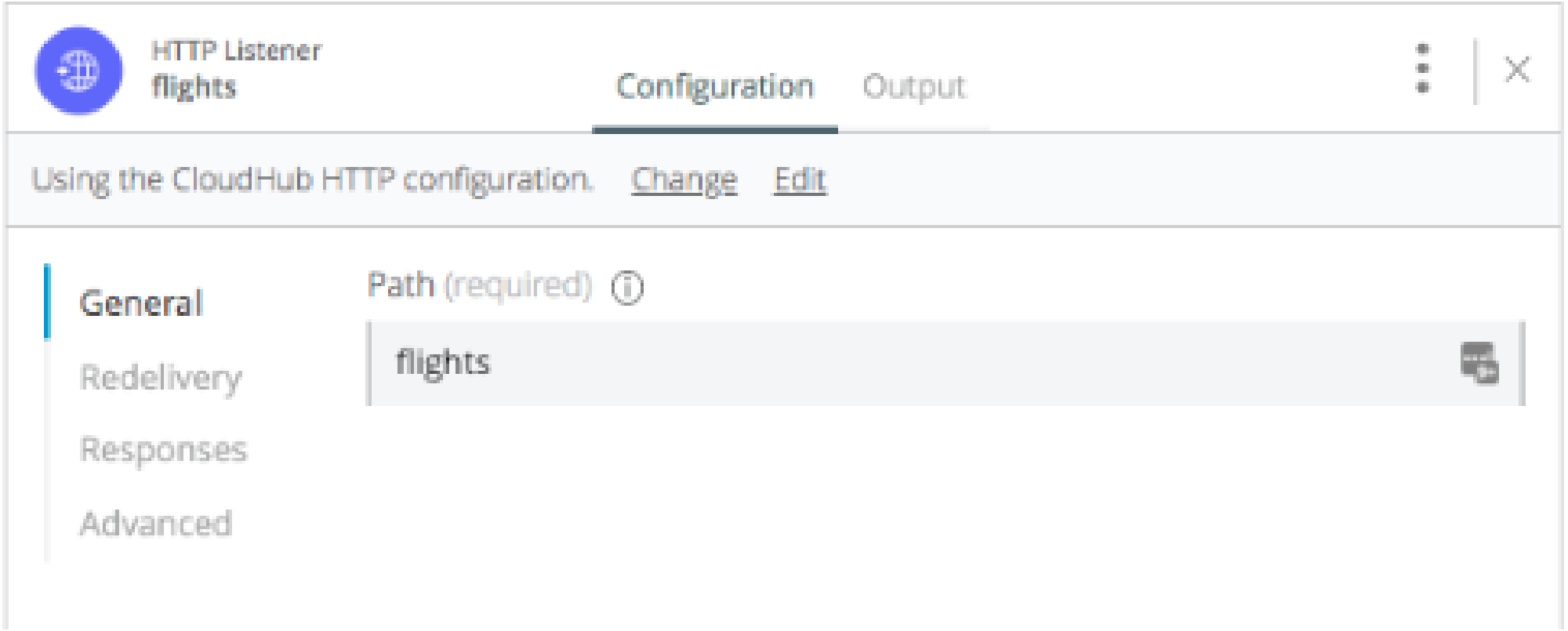


16. Click the Edit link for the CloudHub HTTP configuration.

17. In the HTTP Listener Configuration dialog box, review the information and click Cancel.

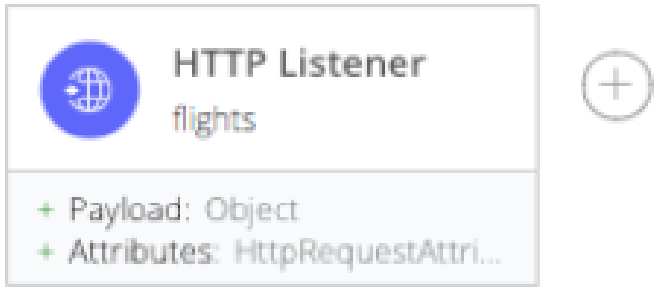


18. In the HTTP Listener dialog box, click the close button in the upper-right corner.

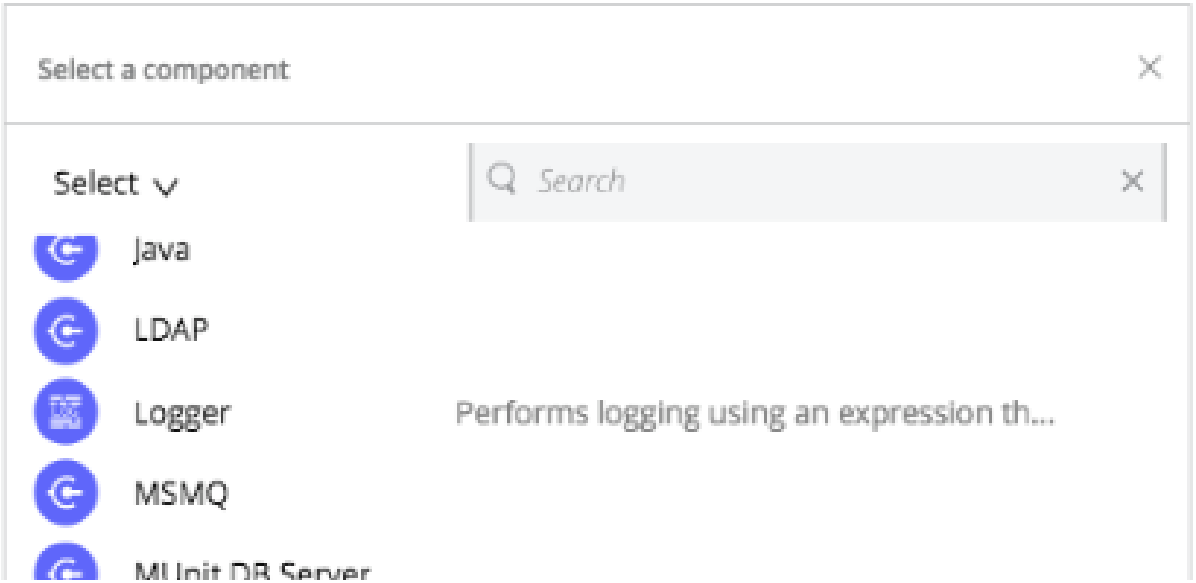


Add a Logger

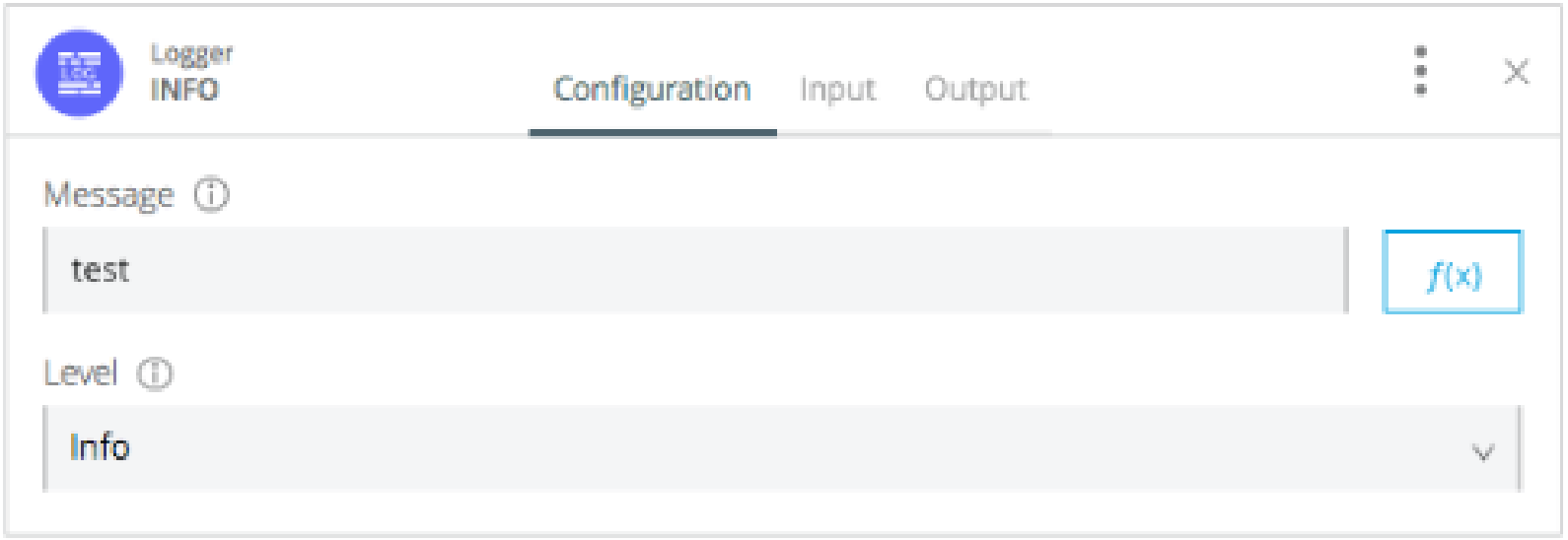
19. Click the add button next to the HTTP Listener card.



20. In the Select a component dialog box, select Logger.



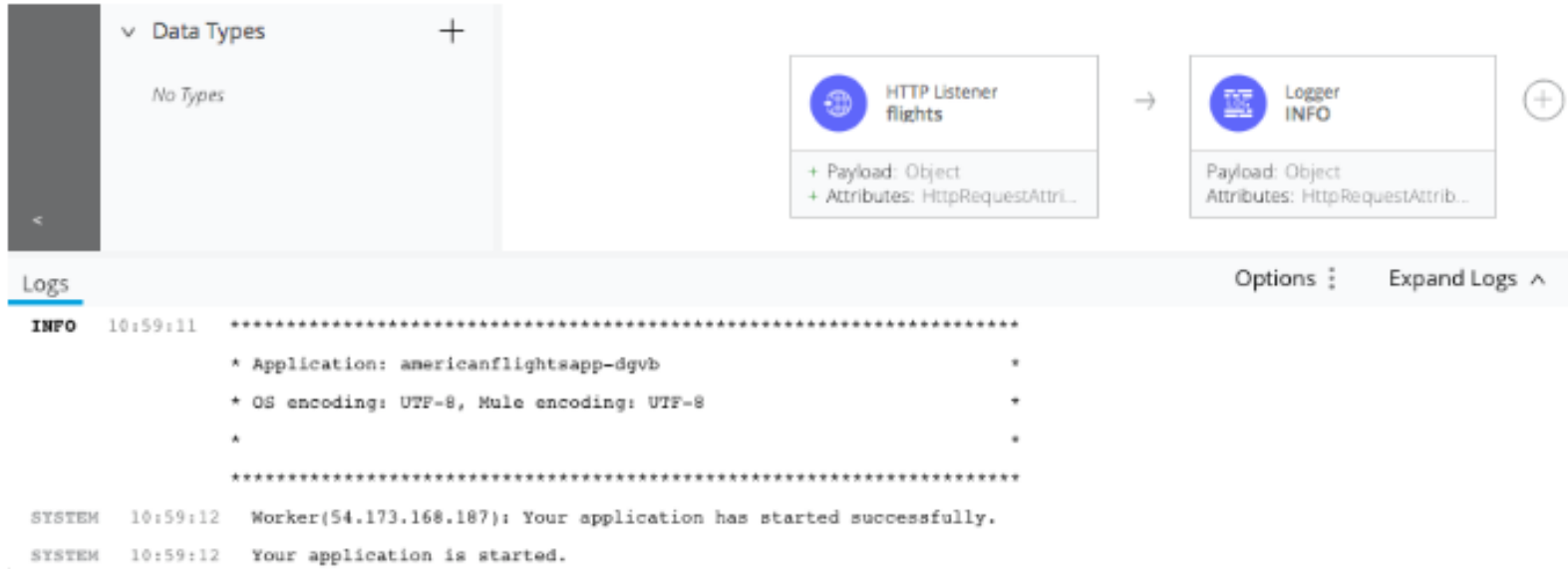
21. In the Logger dialog box, set the message to test.



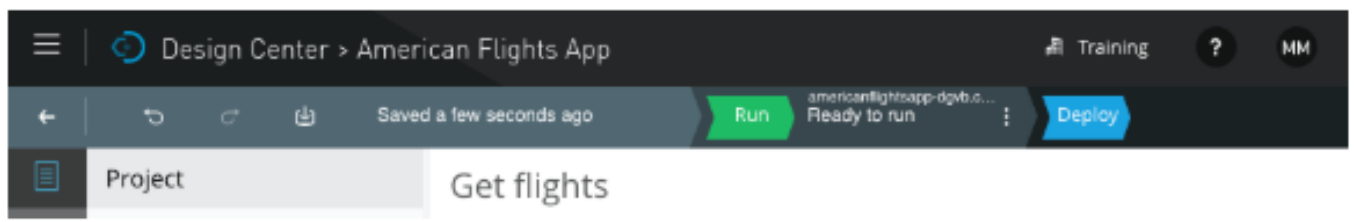
22. Close the card.
23. Notice that there are gray lines across the middle of both cards.

Deploy the application

24. Click the Logs tab located in the lower-left corner of the window; you should see that your application is already started.



25. Locate the application status in the main menu bar; it should say Ready to run.

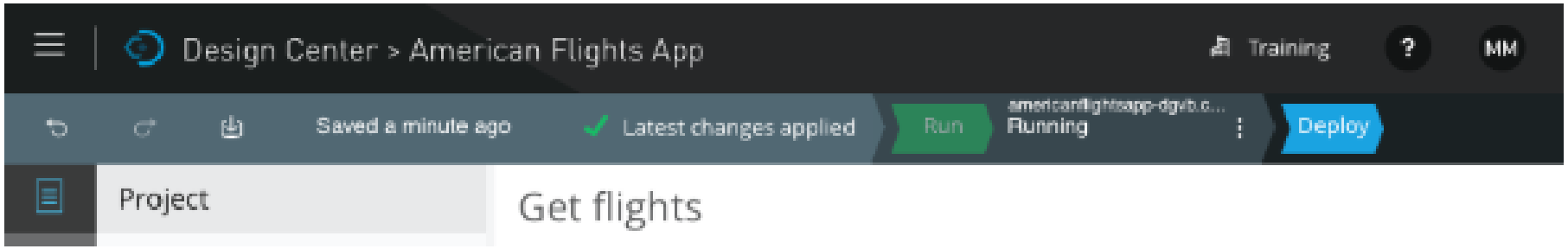


26. Look at the generated URL for the application.

    Note: The application name is appended with a four-letter suffix to guarantee that it is unique across all applications on CloudHub.

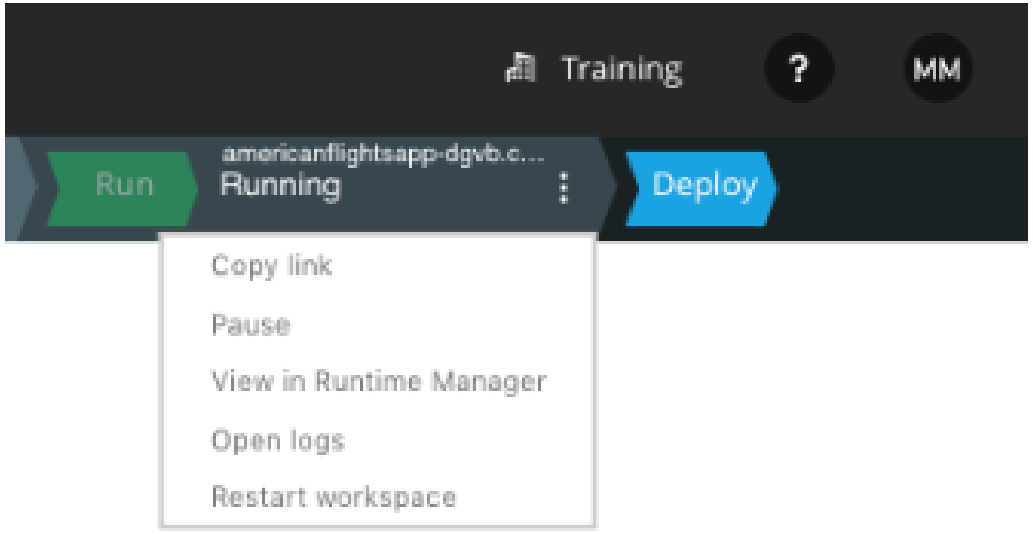
27. Click the Run button.

28. Watch the application status; it should change from Ready to run to Starting application to Running.



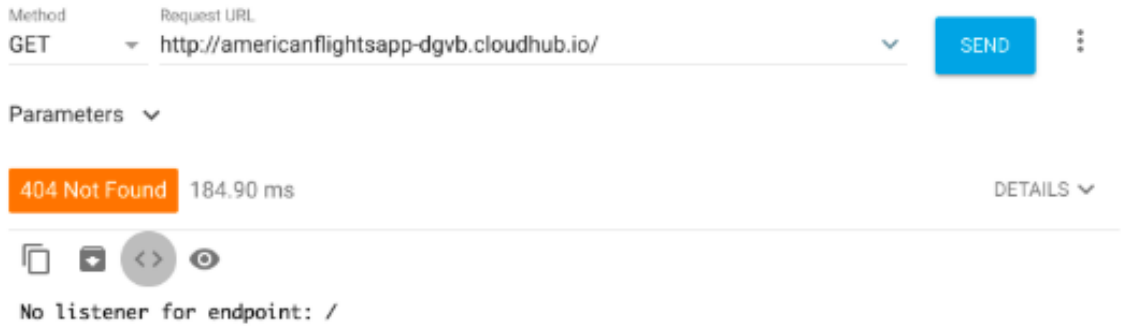
Note: If your application fails to run, look at the messages in the Logs panel. If there is no message a the options menu in the application status area and selecting Restart workspace.

29. Click the options menu in the application status area and select Copy link.

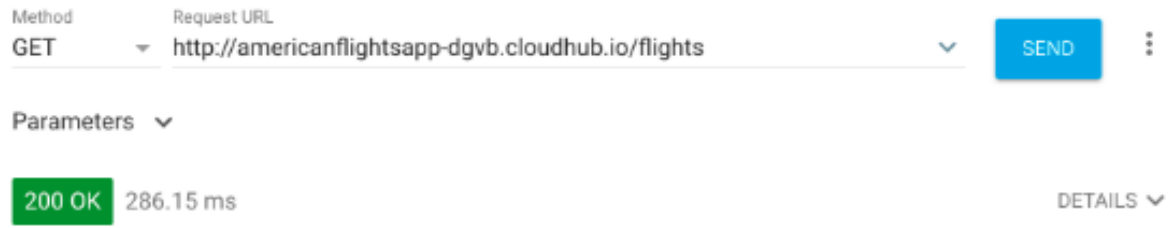


Test the application

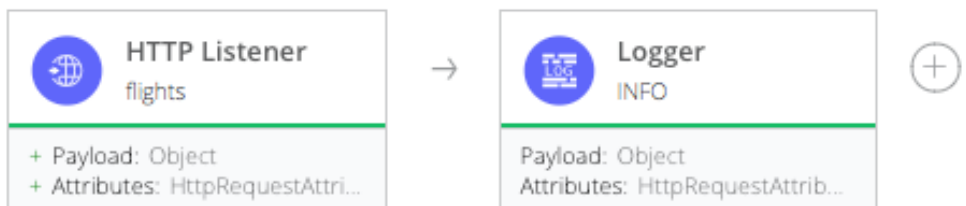
29. Return to Advanced REST Client, paste the copied link, and click Send; you should get a 404 Not Found status with a No listener for endpoint



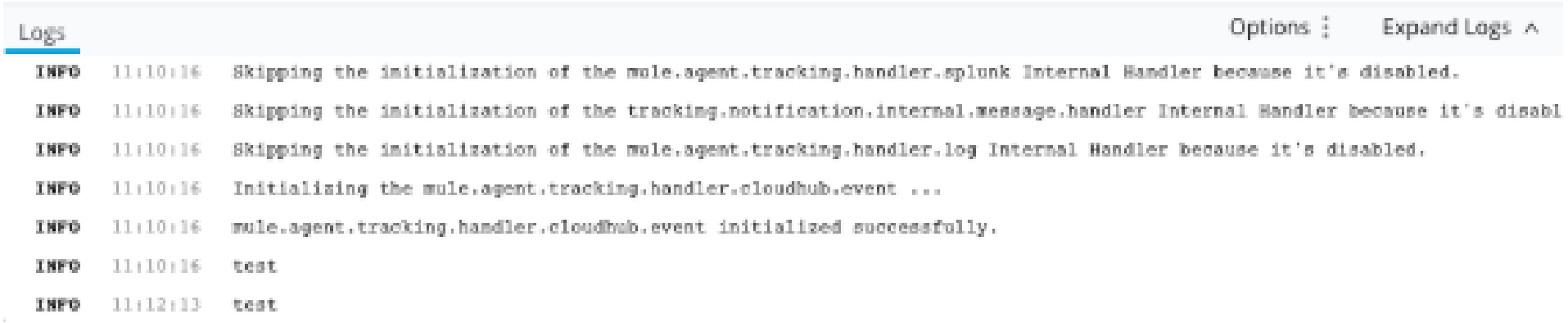
30. Add /flights to the path and click Send; you should get a 200 response with no body.



31. Click Send again to make a second request.
32. Return to flow designer.
33. Notice that there are now green lines across both cards.

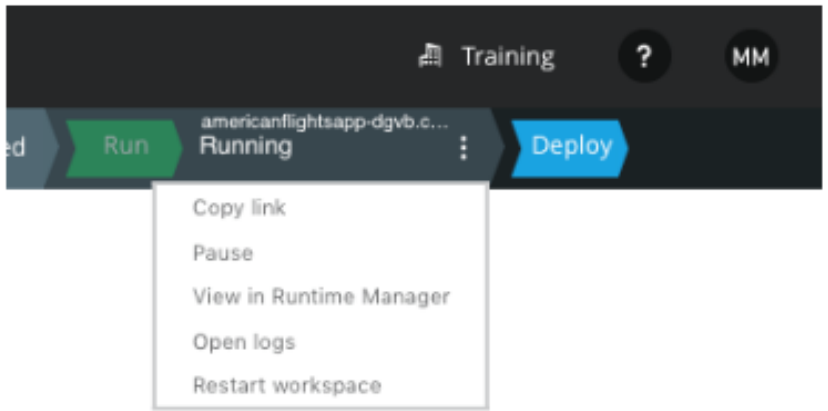


34. Look at the logs; you should see your Logger message displayed twice.

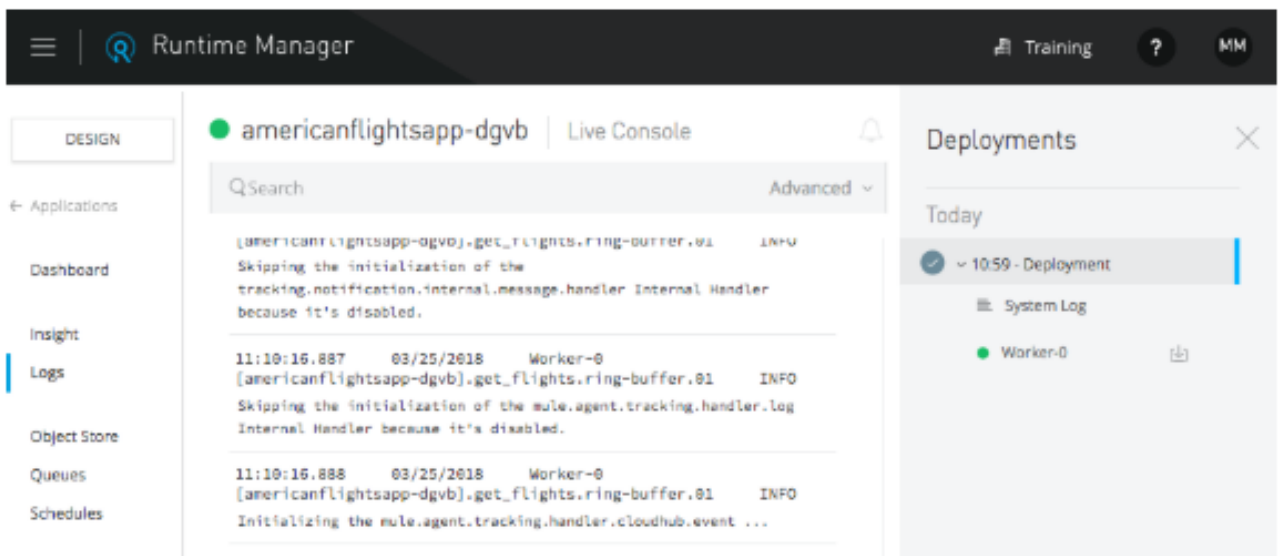


View the application in Runtime Manager

35. Click the options menu in the application status area and select View in Runtime Manager; Runtime Manager should open in a new tab.

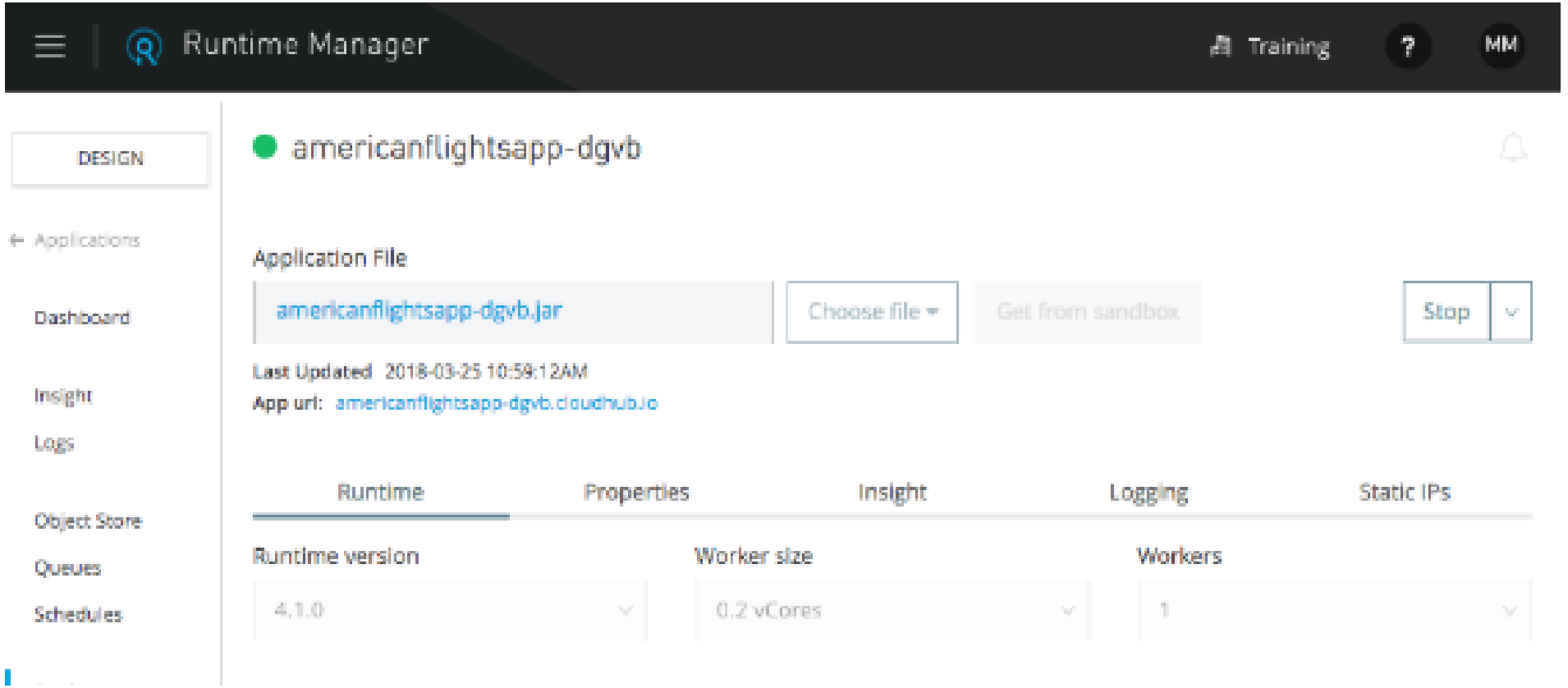


36. In the new browser tab that opens with Runtime Manager, review the application log file; you should see your test log messages.



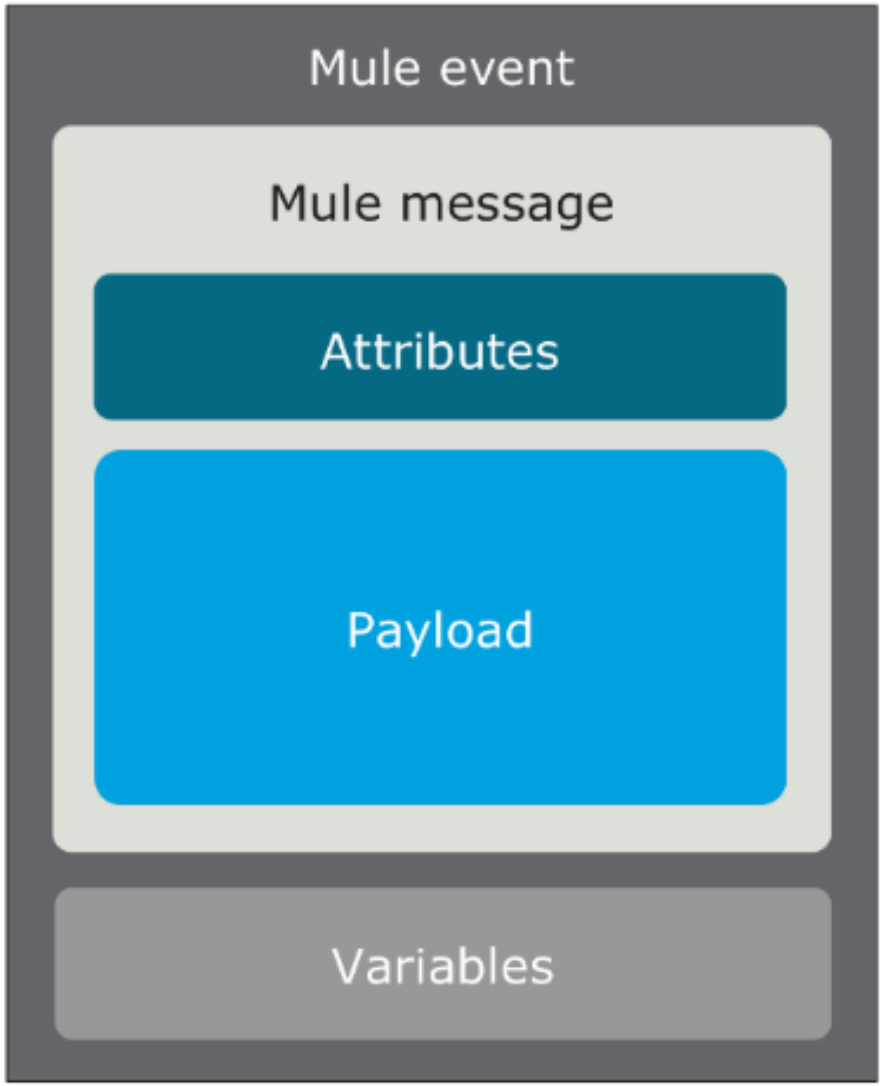
38. Review the settings page and locate the following information for the application:

- To which environment it was deployed
- To what type of Mule runtime it was deployed
- To what size worker it was deployed



Accessing, querying, and transforming data

Accessing and modifying Mule 4 event data



The data that passes through flows in the app

Metadata contained in the message header

The core info of the message - the data the app processes

Metadata for the Mule event - can be defined and referenced in the app processing the event

All contents © MuleSoft Inc.

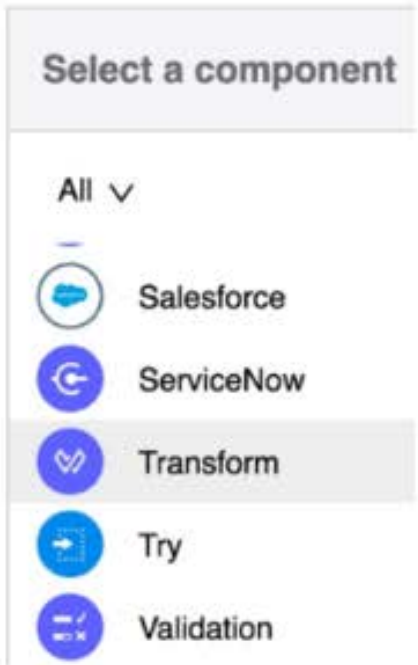
Transforming data with DataWeave

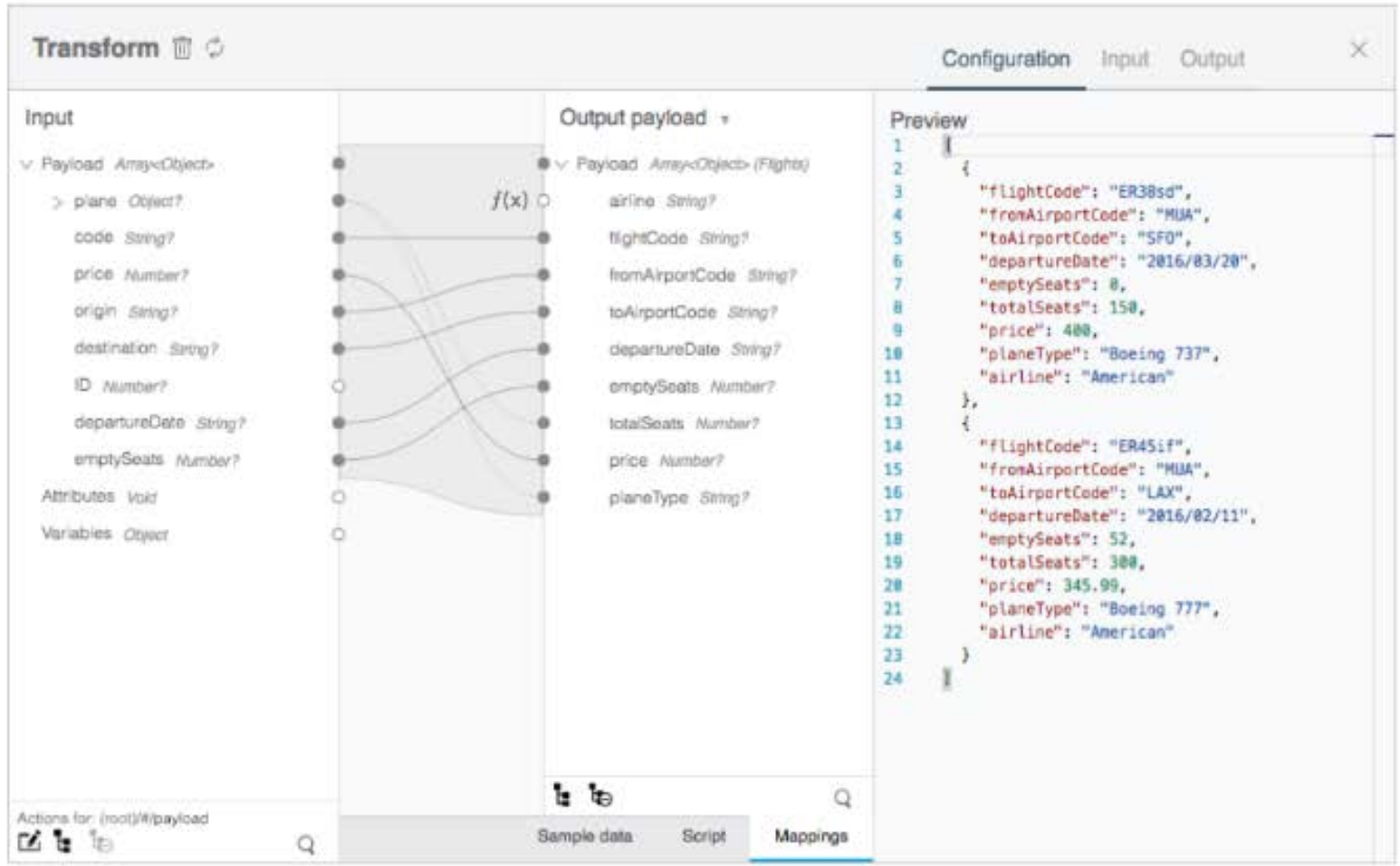
- DataWeave 2.0 is the expression language for Mule to access, query, and transform Mule 4 event data
- A JSON-like language that's built just for data query and transformation use cases
  - Full-featured and fully native framework
- Fully integrated with flow designer (and Anypoint Studio)
  - Graphical interface with payload-aware development

All contents © MuleSoft Inc.

The Transform component

- Has input, output, and preview sections with both drag-and-drop and script editors

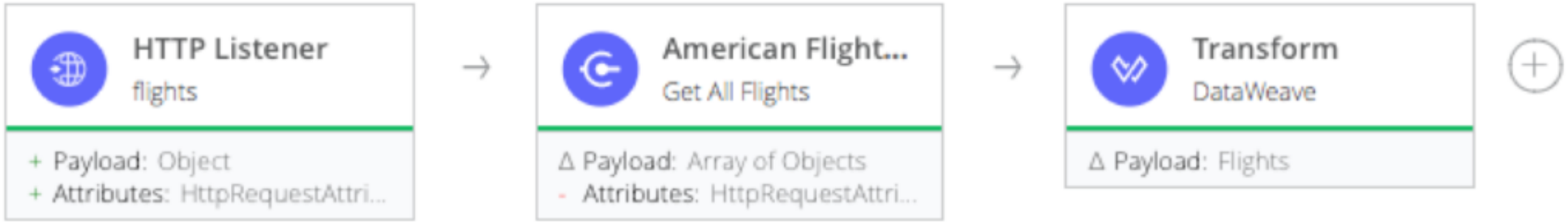




All contents © MuleSoft Inc.

Walkthrough 2-3: Create an integration application with flow designer that consumes an API

- Examine Mule event data for calls to an application
- Use the American Flights API in Anypoint Exchange to get all flights
- Transform data returned from an API to another format

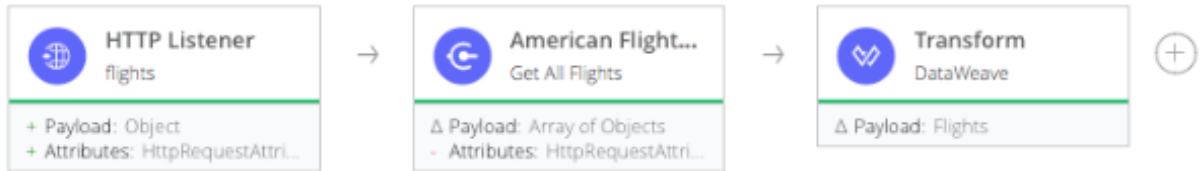


All contents © MuleSoft Inc.

Walkthrough 2-3: Create an integration application with flow designer that consumes an API

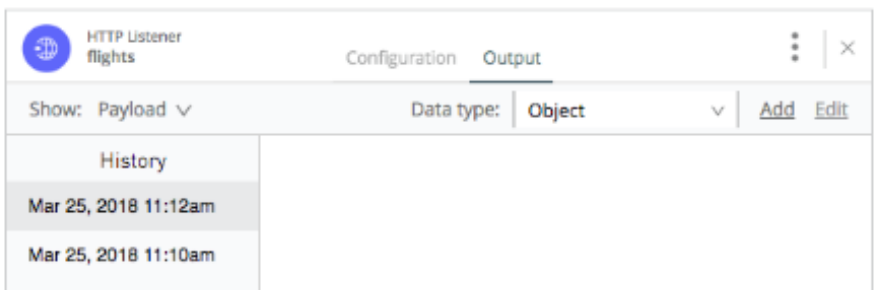
In this walkthrough, you build an integration application to consume an API from Anypoint Exchange. You will:

- Examine Mule event data for calls to an application.
- Use the Training: American Flights API in Anypoint Exchange to get all flights.
- Transform data returned from an API to another format.



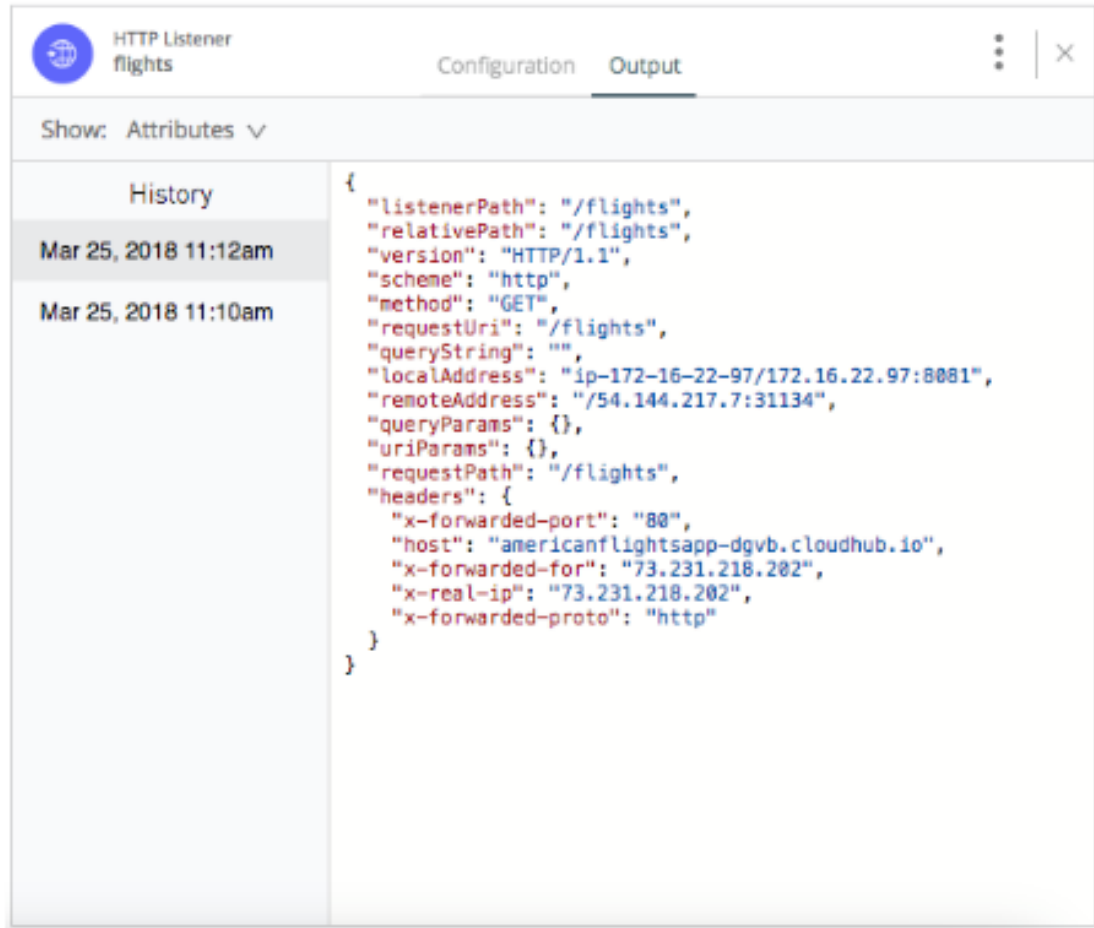
Review Mule event data for the calls to the application

1. Return to the American Flights App in flow designer.
2. Click the title bar of the Logs panel to close it.
3. Expand the HTTP Listener card.
4. Select the Output tab.
5. Locate the Show drop-down menu that currently has Payload selected.



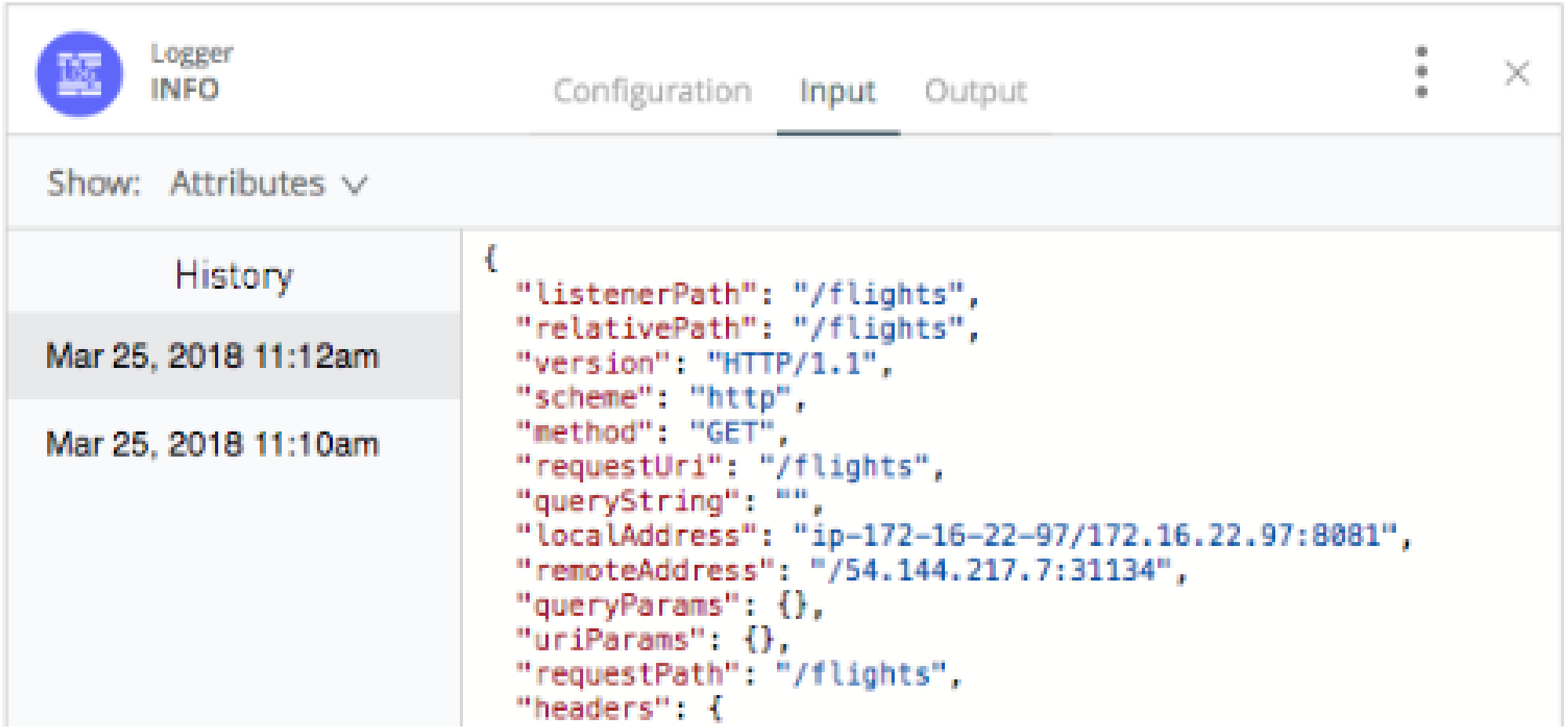
6. Locate your two calls to the application in the History panel; there should be no message payload for either call.
7. Change the Show drop-down menu to Attributes.

8. Review the attributes for the Mule event leaving the HTTP Listener processor.



9. Close the card.
10. Open the Logger card.
11. Select the Input tab.

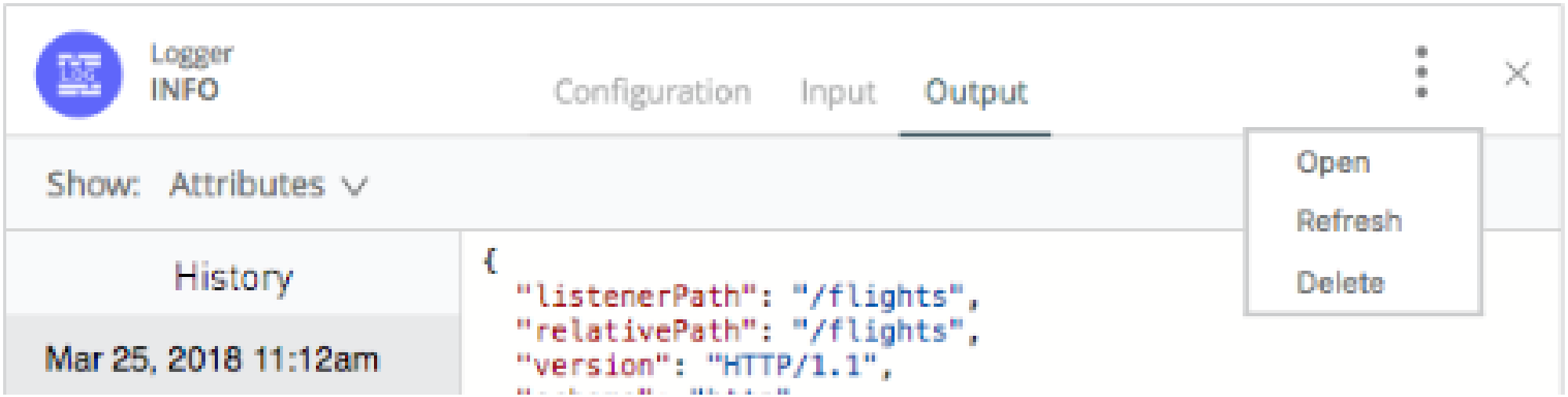
12. Review the payload and attributes values for the two calls.



13. Select the Output tab and review the payload and attributes values for the calls.

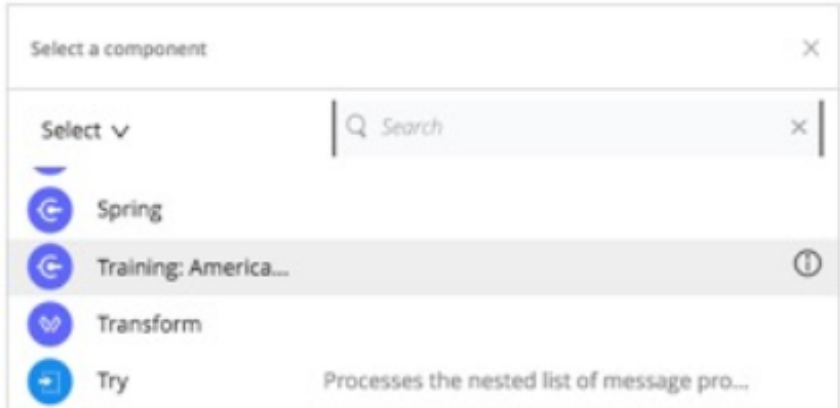
Delete a card

14. Click the options menu for the card and select Delete.

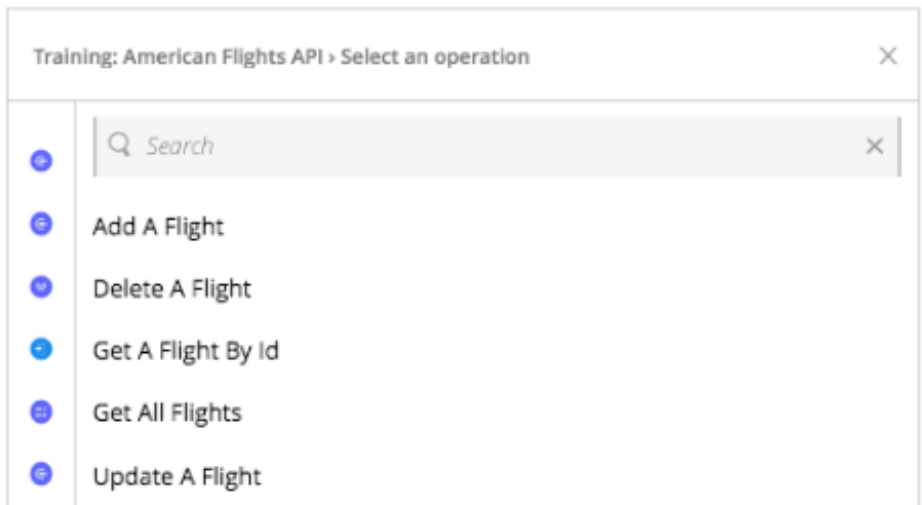


Use the American Flights API in Anypoint Exchange to get all flights

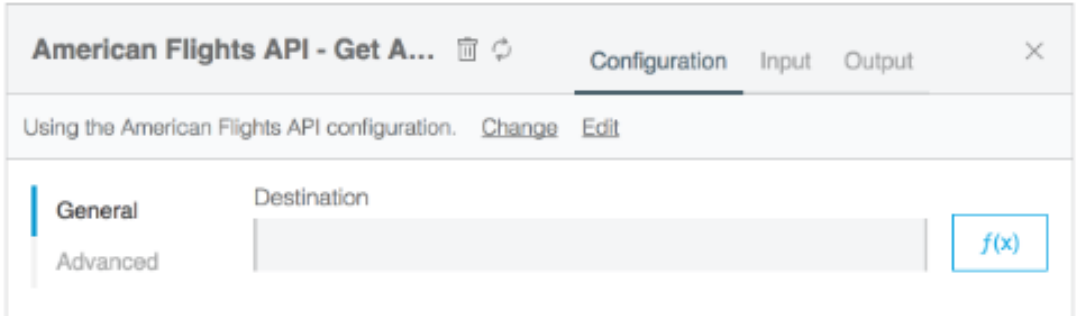
15. Click the Add button next to the HTTP Listener card.
16. In the Select a component dialog box, select the Training: American Flights API.



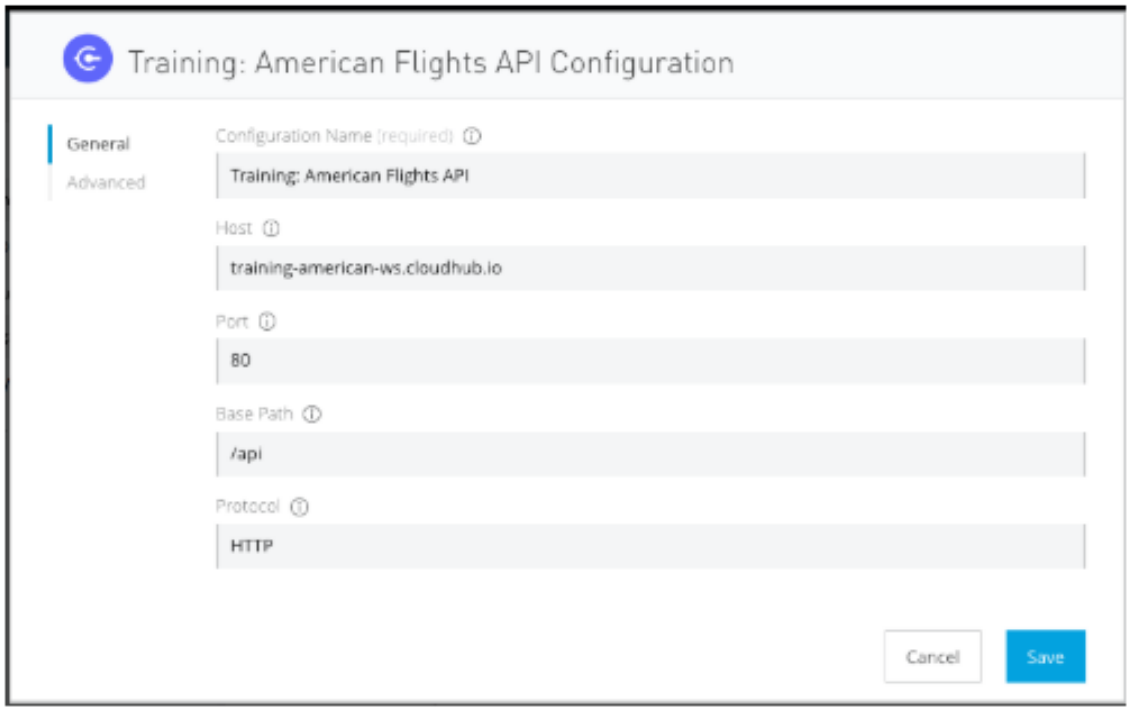
17. In the Training: American Flights API > Select an operation dialog box, select Get All Flights.



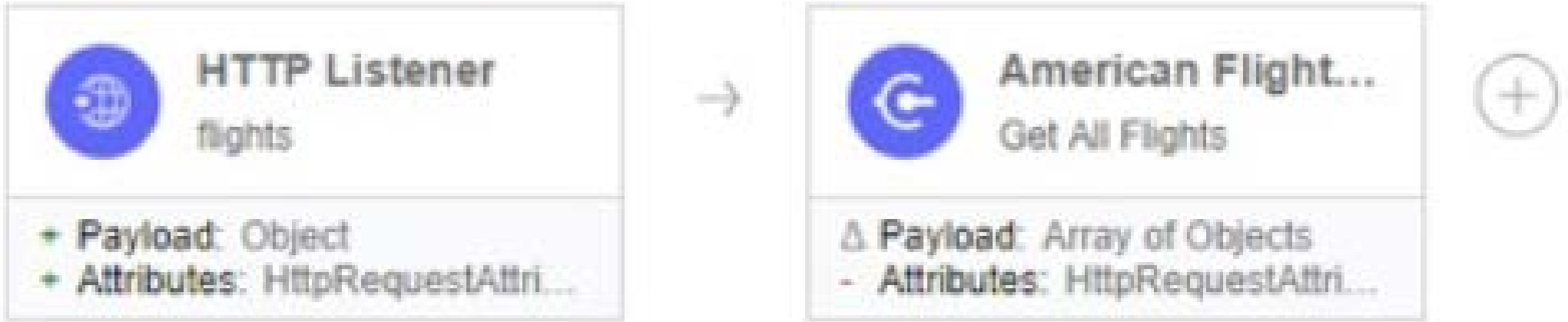
18. In the Get All Flights dialog box, click the Edit link for the Training: American Flights API configuration.



19. Review the information and click Cancel.



20. Close the American Flights API card.



21. Click the Run button in the main menu bar.
22. Wait until the application is running.

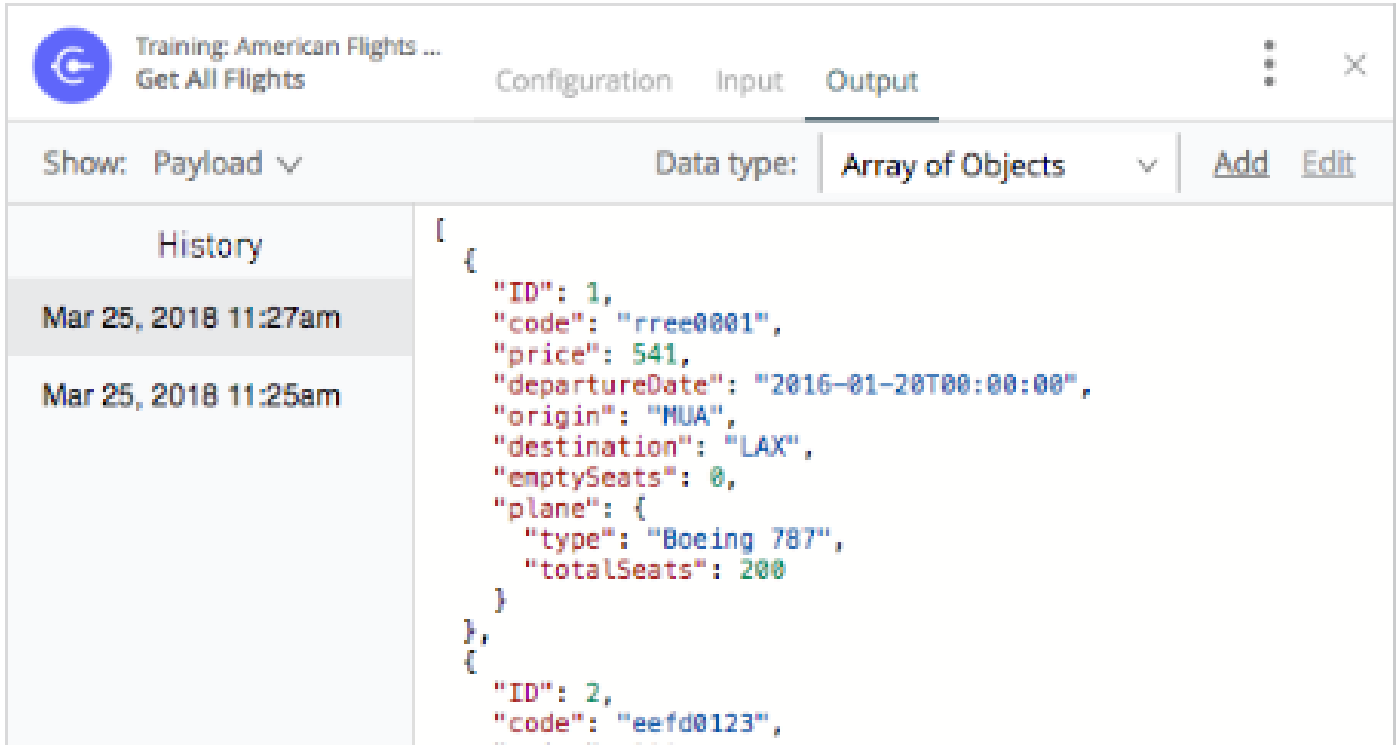
Test the application

23. Return to Advanced REST Client and click Send; you should see flight data.

24. Click Send again to make a second request.

Review Mule event data

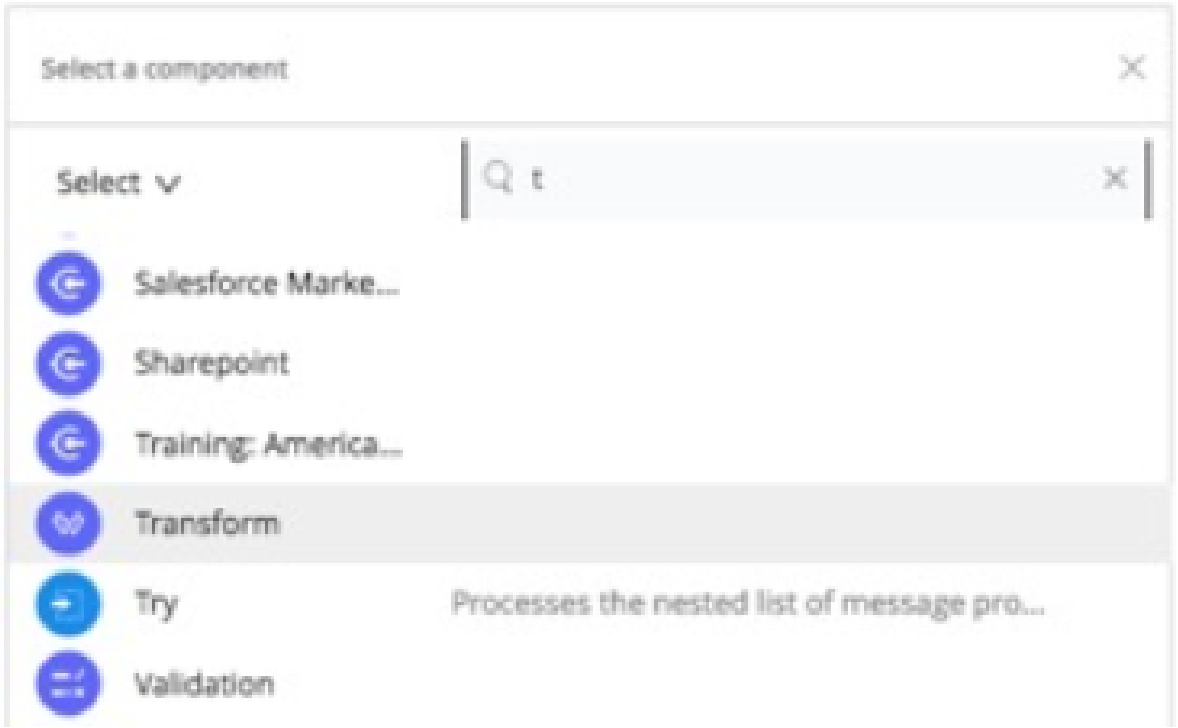
25. Return to flow designer and open the American Flights API card.
26. Select the Input tab and examine the Mule event data.
27. Select the Output tab; you should see payload data.



28. Close the card.

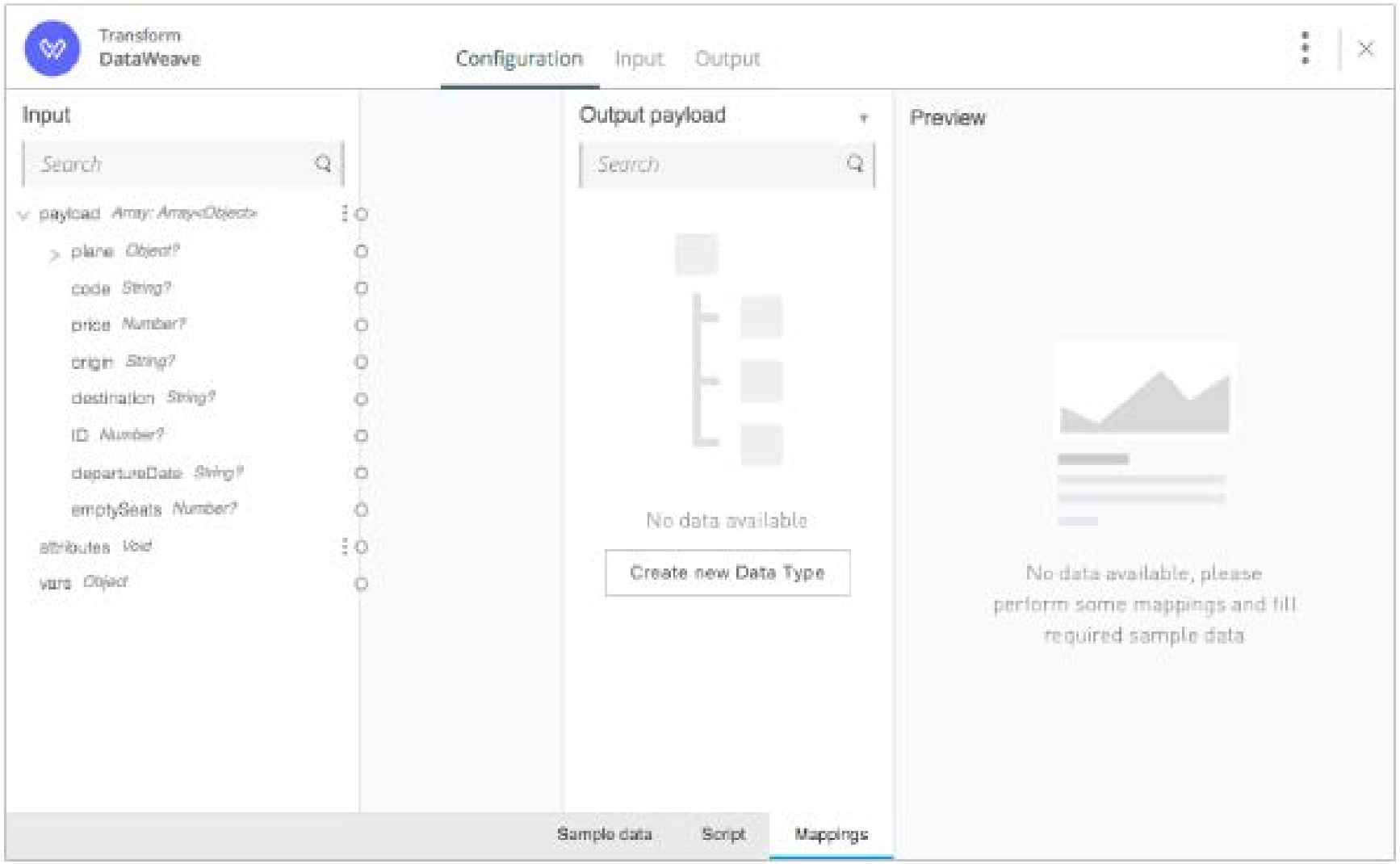
Add and configure a component to transform the data

29. Click the add button in the flow.
30. In the Select a component dialog box, select Transform.

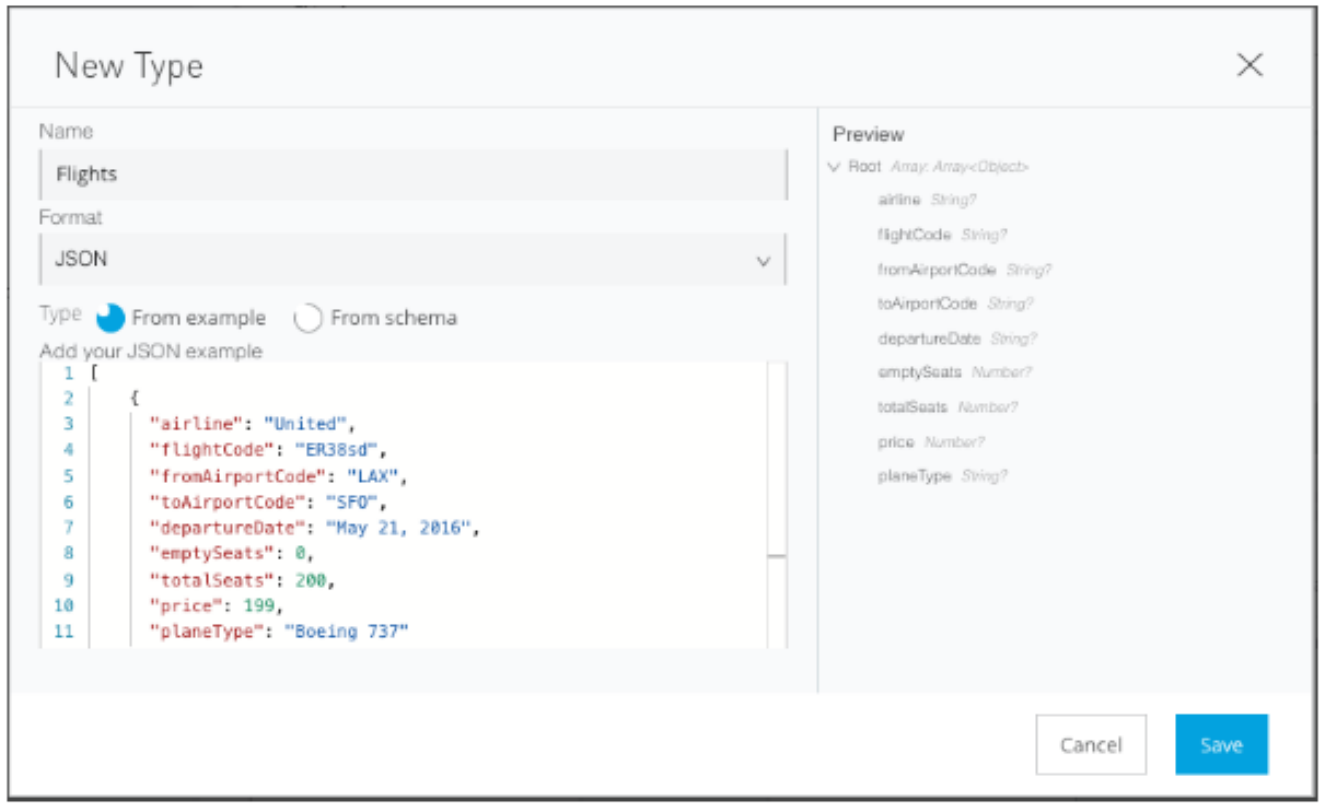


31. In the Transform card, look at the Mule event structure in the input section.

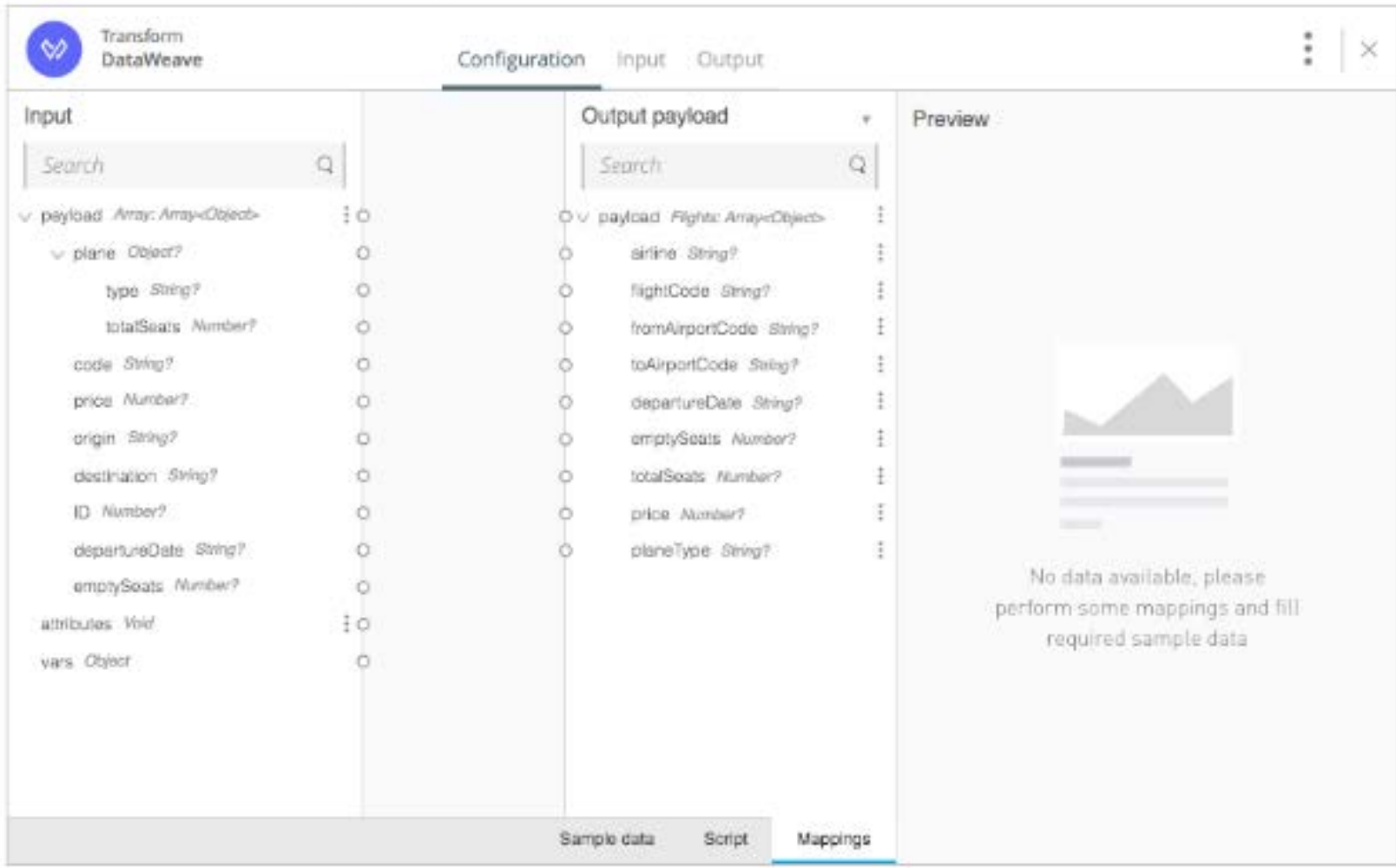
32. In the output section, click the Create new Data Type button.



33. In the New Type dialog box, set the following values:
    - Name: Flights
    - Format: JSON
    - Type: From example
34. In the computer's file explorer, return to the student files folder and locate the flights-example.json file in the examples folder.
35. Open the file in a text editor and copy the code.
36. Return to flow designer and paste the code in the section to add your JSON example.



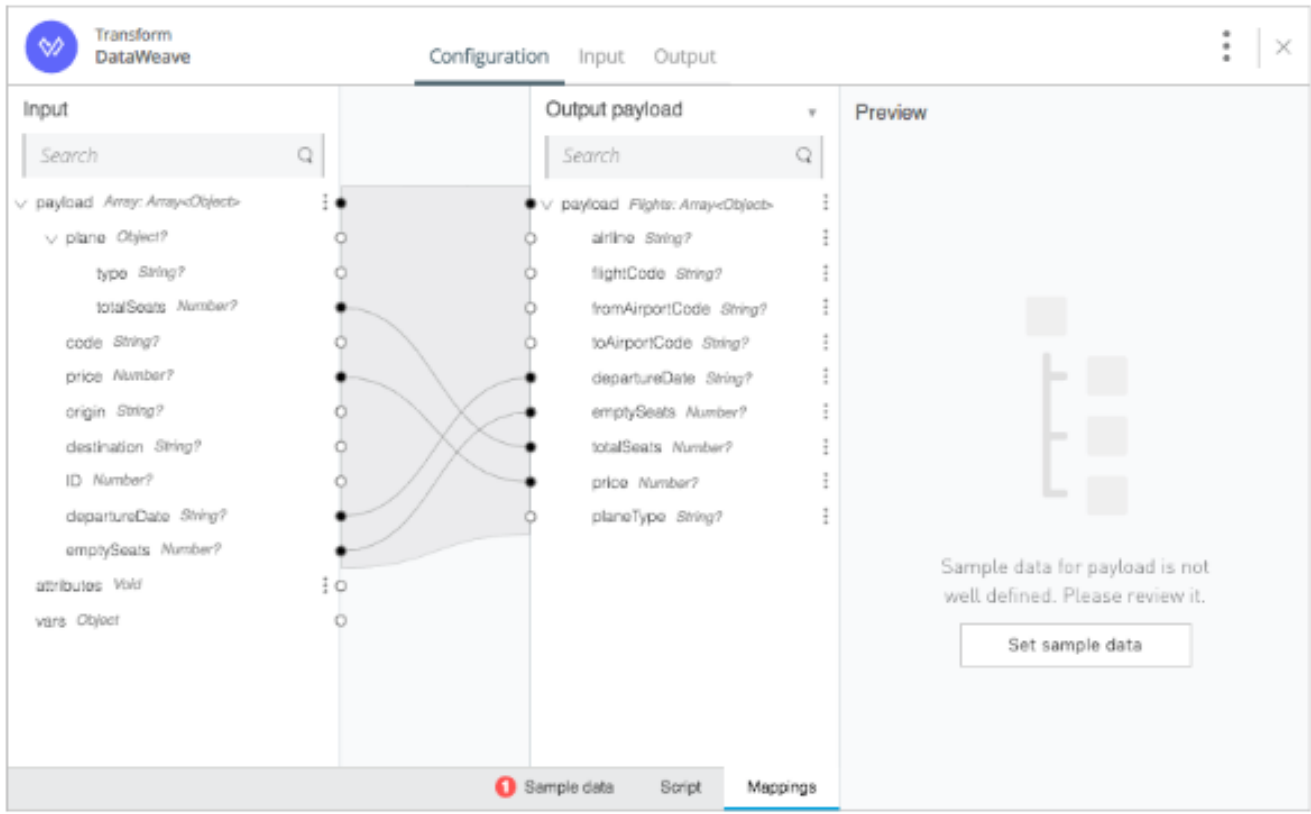
37. Click Save.
38. In the input section, expand the plane object.



Create the transformation

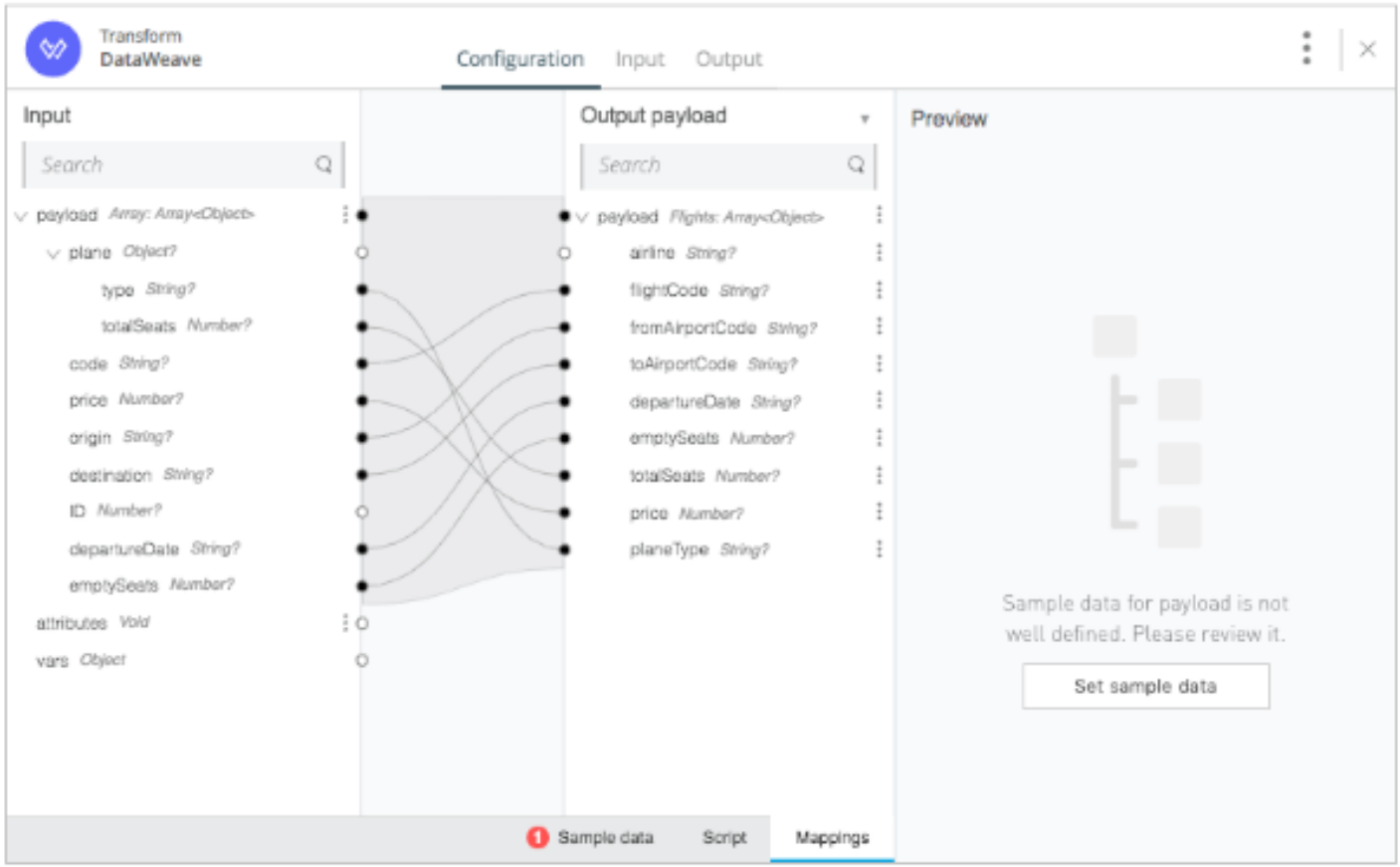
39. Map fields with the same names by dragging them from the input section and dropping them on the corresponding field in the output section.

- price to price
- departureDate to departureDate
- plane > totalSeats to totalSeats
- emptySeats to emptySeats

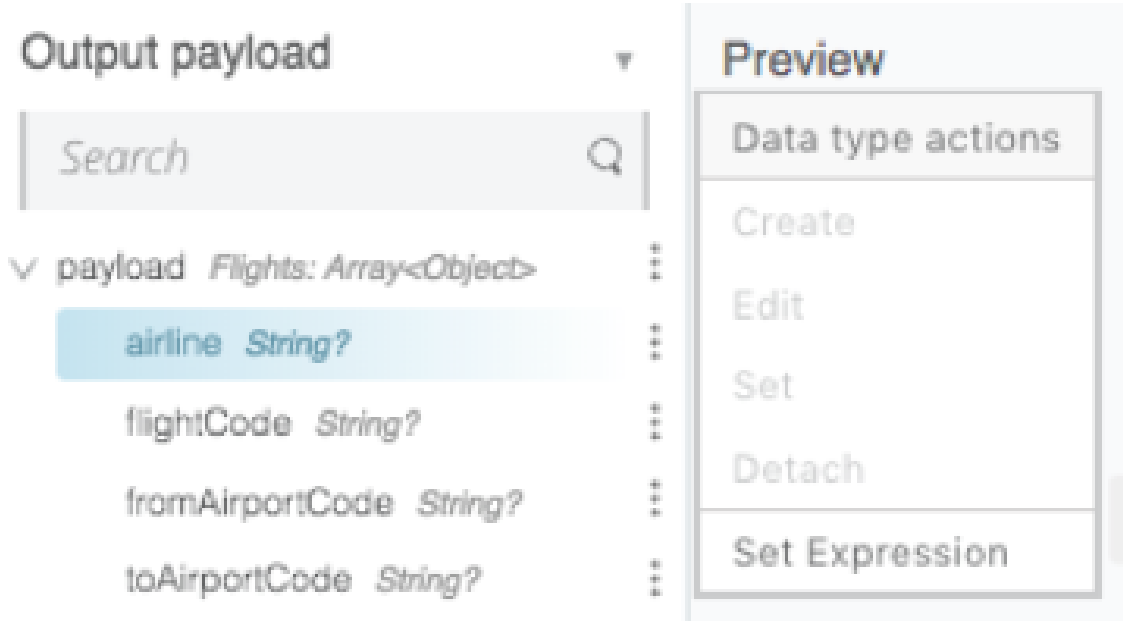


40. Map fields with different names by dragging them from the input section and dropping them on the corresponding field in the output section.

- plane > type to planeType
- code to flightCode
- origin to fromAirport
- destination to toAirport



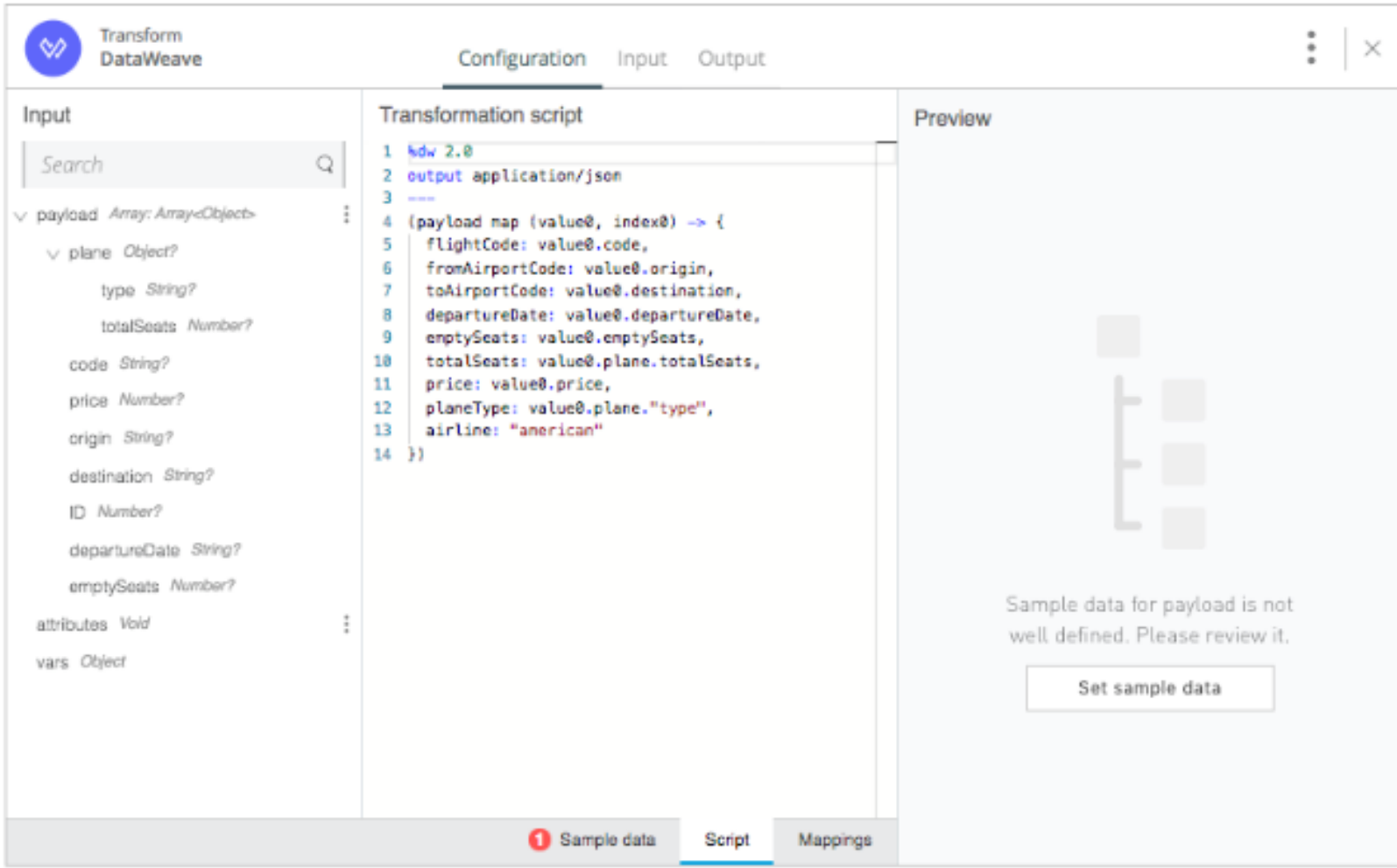
41. In the output section, click the options menu for the airline field and select Set Expression.



42. Change the value from null to "american" and click OK.

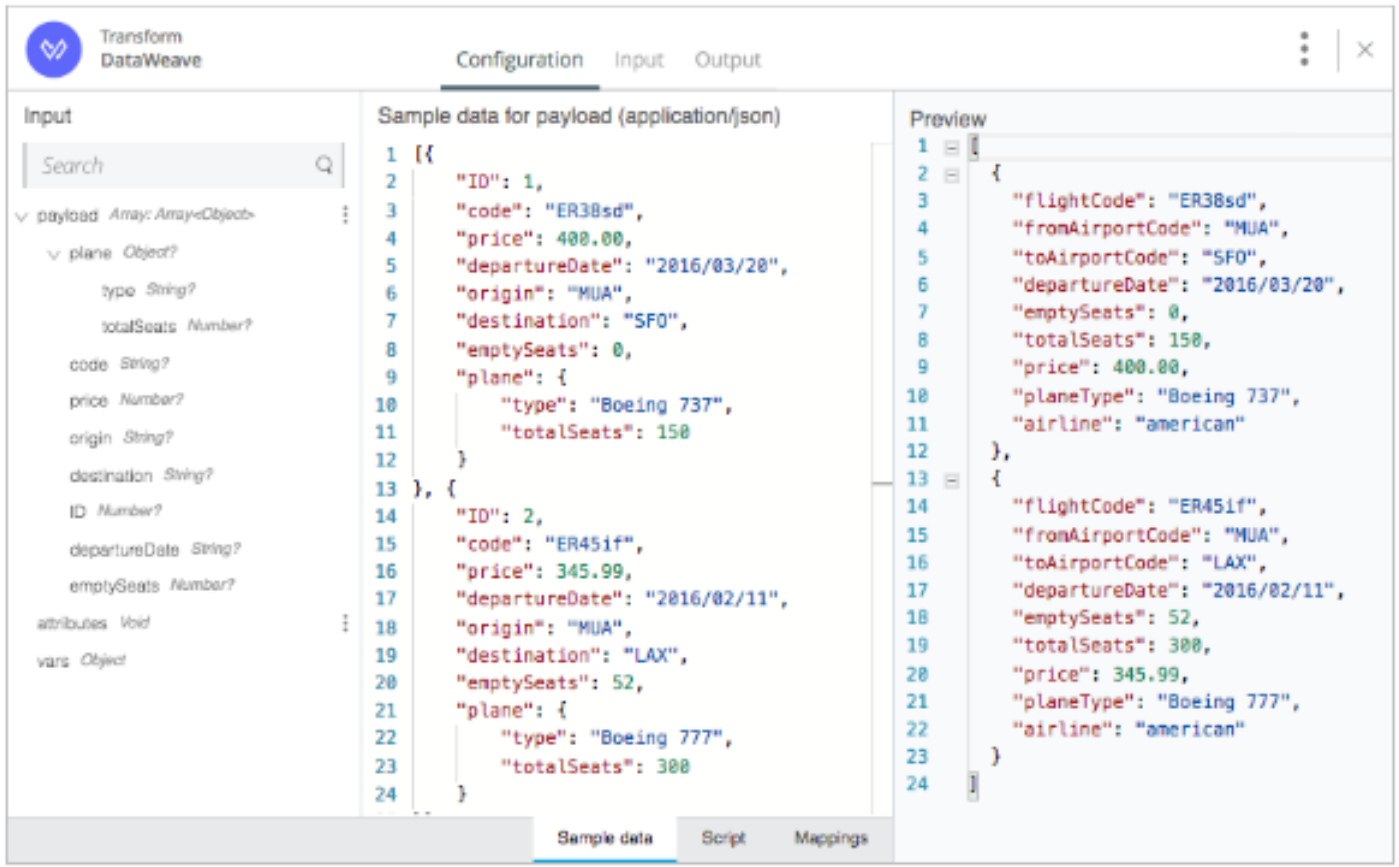
43. Click the Script tab at the bottom of the card; you should see the DataWeave expression for the transformation.

Note: You learn to write DataWeave transformations later in the Development Fundamentals course.

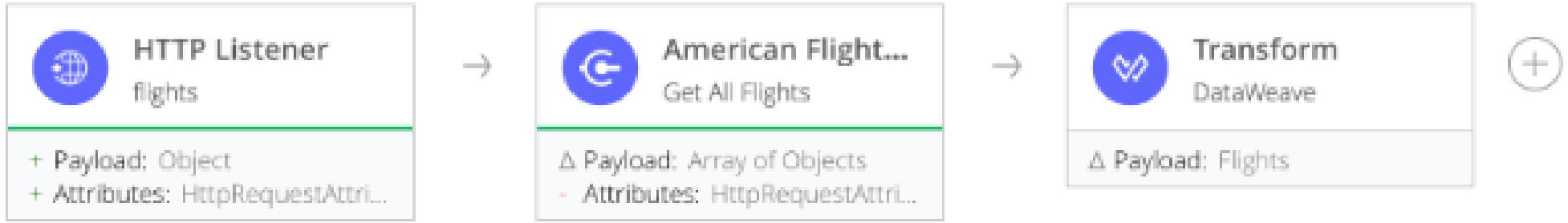


Add sample data

44. Click the Set sample data button in the preview section.
45. In the computer's file explorer, return to the student files folder and locate the american-flights-example.json file in the examples folder.
46. Open the file in a text editor and copy the code.
47. Return to flow designer and paste the code in the sample data for payload section.

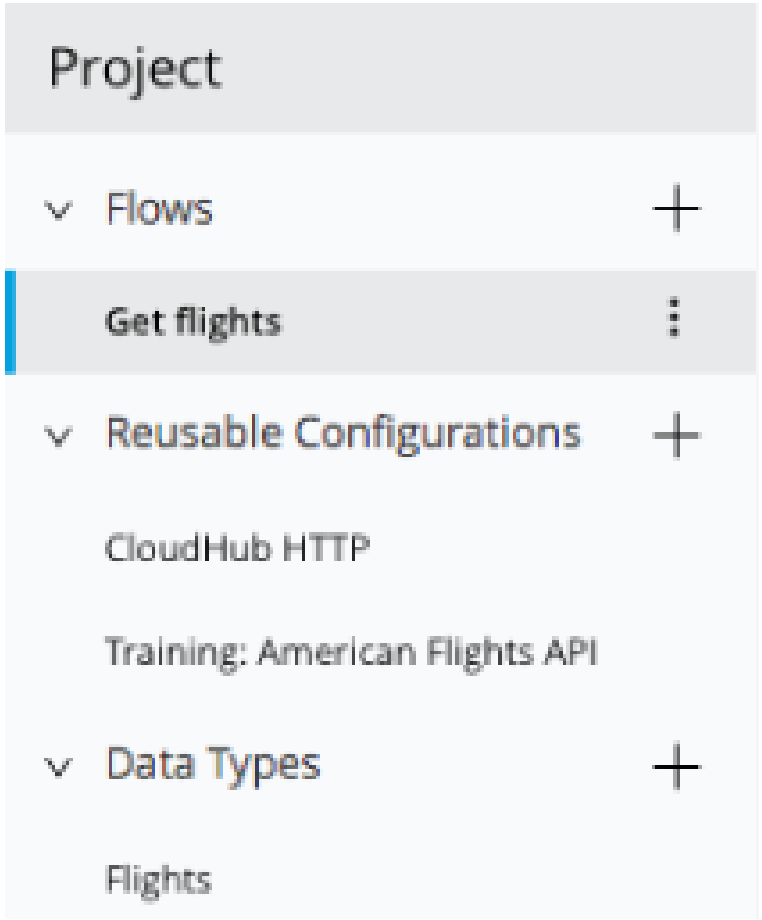


48. Look at the preview section, you should see a sample response for the transformation.
49. Close the card.



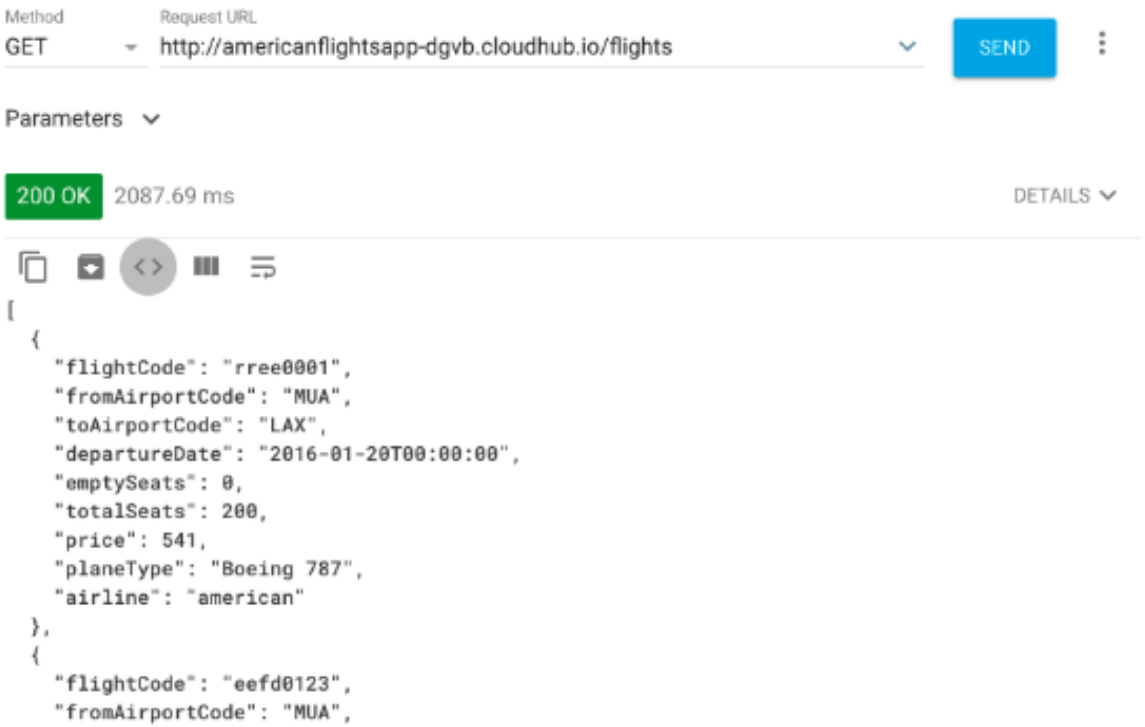
Locate the data type and configuration definitions

50. Locate the connector configuration and the new Flights data type in the project explorer.



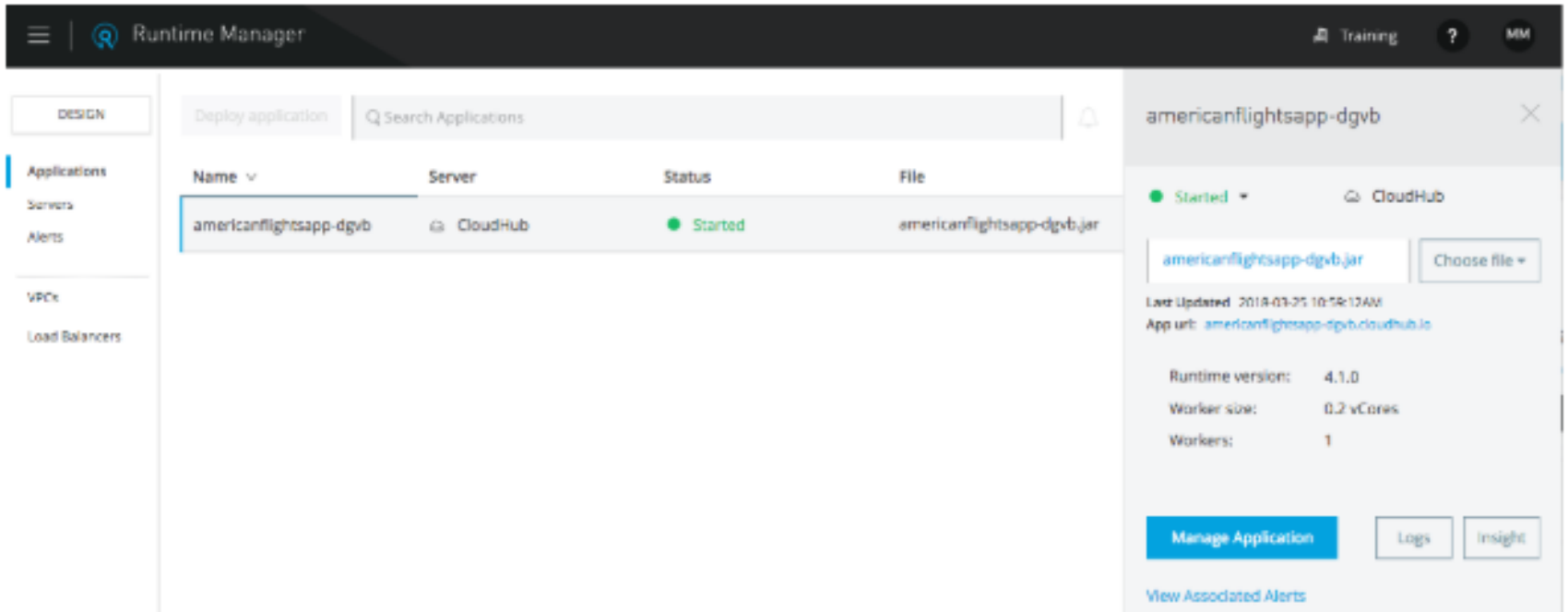
Test the application

51. Run the project.
52. Return to Advanced REST Client and click Send to make another request to http://americanflightsapp-xxxx.cloudhub.io/flights; you should see all the flight data as JSON again but now with a different structure.



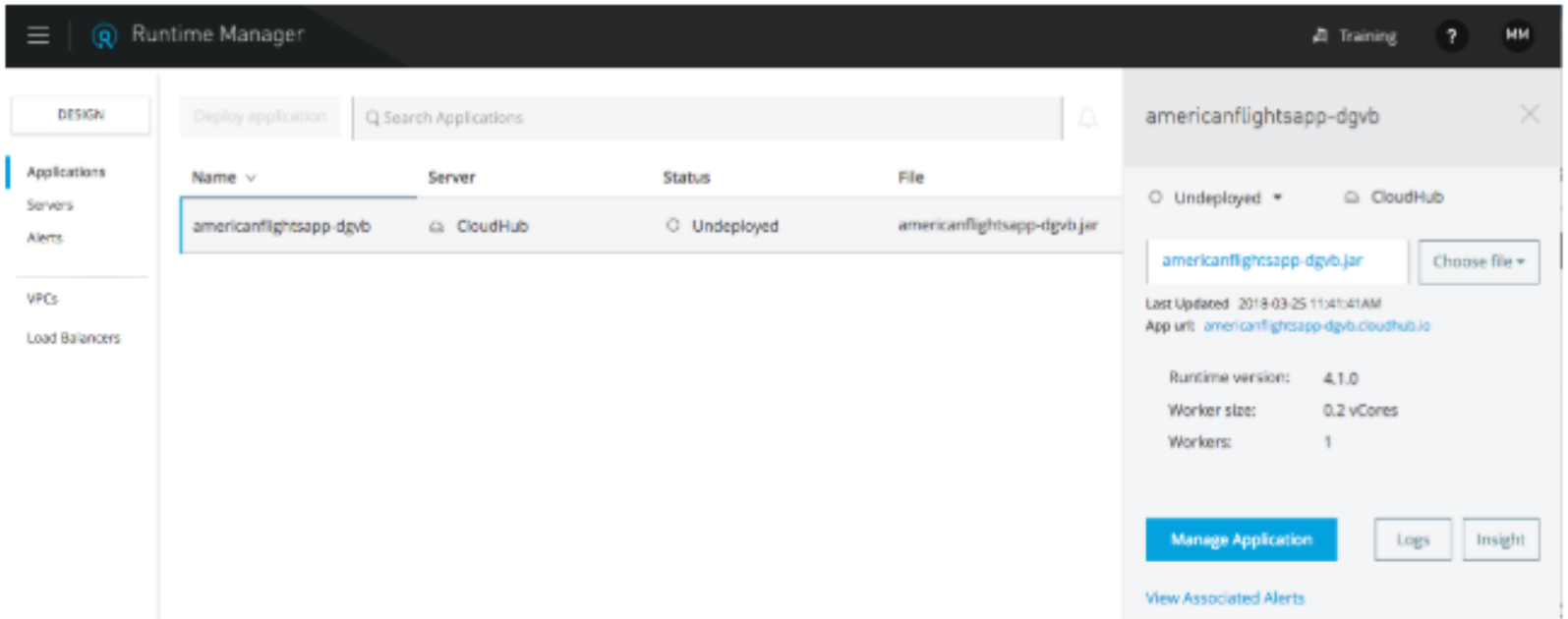
Stop the application

53. Return to Runtime Manager.
54. In the left-side navigation, click Applications.
55. Select the row with your application; you should see information about the application displayed on the right side of the window.

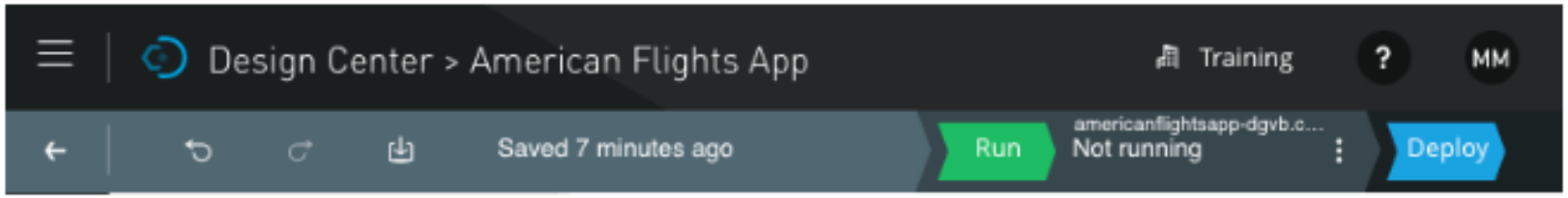


56. Click the drop-down menu button next to Started and select Stop; the status should change to Undeployed.

    Note: You can deploy it again from flow designer when or if you work on the application again.



57. Close Runtime Manager.
58. Return to flow designer; you should see the application is not running.

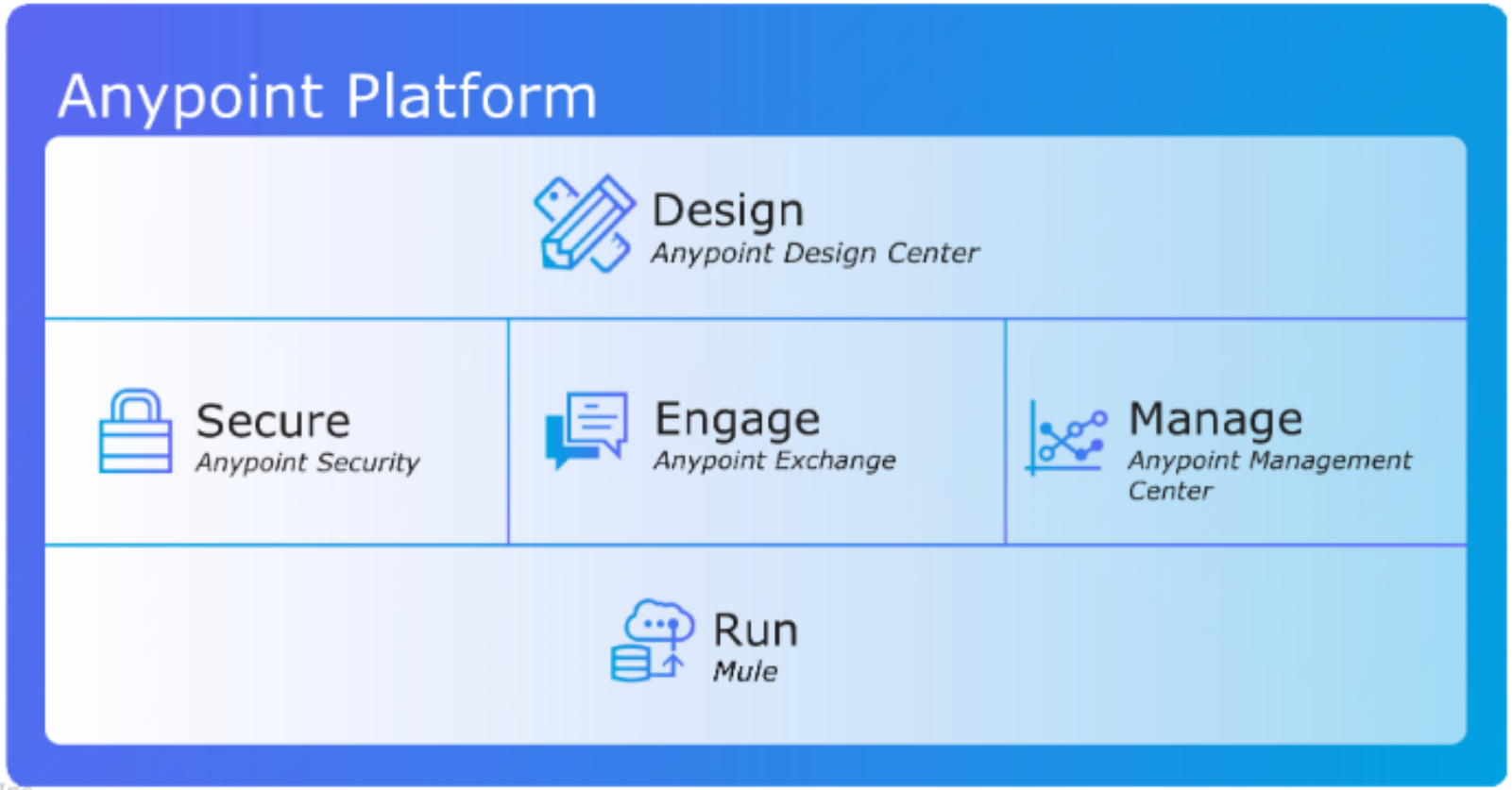


59. Return to Design Center.

Summary

Summary: Anypoint Platform

- **Anypoint Platform** is a unified, hybrid integration platform that creates a seamless **application network** of apps, data, and devices with **API-led connectivity**



All contents © MuleSoft Inc.

Summary

- Use **Anypoint Exchange** as a central repository for assets so they can be discovered and reused
  - Populate it with everything you need to build your integration projects
- Use **flow designer** to build integration applications
  - These are Mule 4 applications that are deployed to a Mule runtime
  - To learn more, take the 1-day *Anypoint Platform: Flow Design* course
- **Mule runtimes** can be MuleSoft-hosted in the cloud (CloudHub) or customer-hosted in the cloud or on-prem
- **DataWeave 2.0** is the expression language for Mule to access, query, and transform Mule 4 event data

All contents © MuleSoft Inc.